NON-LINEAR SUBDIVISION OF UNIVARIATE SIGNALS AND DISCRETE SURFACES

THÈSE N^{O} 2815 (2003)

PRÉSENTÉE À LA FACULTÉ SCIENCES ET TECHNIQUES DE L'INGÉNIEUR

Institut de traitement des signaux

SECTION D'ÉLECTRICITÉ

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES TECHNIQUES

PAR

Nicolas ASPERT

ingénieur diplômé de l'ENSIE Grenoble, France
et de nationalité française

acceptée sur proposition du jury:

Prof. T. Ebrahimi, directeur de thèse
Prof. N. Garcia Santos, rapporteur
Prof. R. Leonardi, rapporteur
Prof. D. Thalmann, rapporteur
Prof. P. Vandergheynst, rapporteur

Lausanne, EPFL
2003

Remerciements

Je tiens à remercier tout d'abord mon directeur de thèse, le Prof. Touradj Ebrahimi, pour son soutien et ses conseils, ainsi que pour la liberté offerte quant au choix du sujet de ma thèse. Ce travail a aussi été le fruit de nombreuses discussions et réflexions menées conjointement avec le Prof. Pierre Vandergheynst, dont les connaissances mathématiques m'ont été d'un grand secours pour certaines parties de cette thèse, mais avec qui j'ai également eu le plaisir de faire quelques activités extra-scientifiques fort sympathiques, telles ces mémorables journées de ski du laboratoire ou les "via ferrate" des Dolomites que nous avons parcourues. Je souhaite également remercier le directeur de l'ITS, le Prof. Murat Kunt, pour son soutien et son aide lors de certains moments difficiles, ainsi que les membres de mon jury de thèse pour leurs commentaires et suggestions.

Je remercie tout particulièrement le Prof. Francisco Morán Burgos de l'université de Madrid, pour m'avoir fourni le code source de son programme de compression de modèles 3D qu'il a développé pendant sa thèse. Sans sa collaboration, le chapitre dédié aux applications se serait trouvé fort réduit.

Du fait de la proximité de nos bureaux, mais aussi de celle de nos sujets de thèse, j'ai eu la chance de pouvoir travailler en collaboration avec Diego Santacruz. Je ne compte plus les questions sur les arcanes du C, de Linux, de JPEG 2000, du MQ-coder et beaucoup d'autres sujets, auxquelles il a su me répondre avec patience et précision. Son aide a été plus que précieuse et m'a permis d'apprendre énormément sur de nombreux sujets.

Pour la bonne ambiance qui règne au laboratoire, grâce aux discussions animées pendant les pauses-café, séances de cinéma "open-air" sur la terrasse et autres verres à Sat', merci à tous mes collègues du LTS. Je tiens aussi à remercier particulièrement les secrétaires du laboratoire, Marianne Marion et Isabelle Bezzi pour leur aide indispensable dans les problèmes administratifs.

Parmi ces collègues de travail, il en est un que je tiens à remercier plus particulièrement pour sa double casquette de collègue et violoniste, Olivier Steiger. Les nombreuses heures passées en sa compagnie à jouer du violon ou de l'alto, à l'orchestre, en musique de chambre, ou pendant une des nombreuses répétitions en duo dans les auditoires du département d'électricité m'ont souvent aidé à me changer les idées. J'ai trouvé très appréciable d'avoir un collègue sympathique et toujours enthousiaste pour un peu de musique.

La pratique de la musique a occupé une place importante durant les années passées à l'EPFL, c'est pourquoi je tiens à remercier les personnes avec qui j'ai eu l'occasion de faire de la musique de chambre ainsi que l'équipe de l'orchestre universitaire, tant pour les moments musicaux eux-mêmes que pour les après-répétitions et autres après-concerts. Merci en particulier à Natalia et Martin Hasler, Benjamin Ilschner, Bernard Roder, Johannes Schneider, Jérôme Bonnet et Jean-Marc Bonard pour les longues soirées de musique de chambre, à Tabea Leibbrand et Franziska Lenz pour m'avoir fait découvrir et donné le goût du quatuor à cordes, à Joaquim Esmerado, formidable pianiste avec qui j'ai passé de longues heures à découvrir les sonates de Brahms et Beethoven, et finalement à Camille Gabioud qui m'a vaillamment supporté comme voisin de pupitre à l'orchestre pendant la période difficile que constitue la rédaction d'une thèse, et qui malgré cela a bien voulu passer quelques heures de plus à jouer des duos pour violon et alto avec moi.

Pour (entre autres) les concerts de musique un peu moins classique, les tours en VTT, les courses d'escalade à but botanique (cueillette du génépi), merci à Julien Vallet dont l'enthousiasme et le dynamisme constants sont autant de remèdes à mon penchant naturel vers l'oisiveté.

Enfin, je remercie ma famille, notamment mes parents et mon frère François pour leur soutien, leurs encouragements et leur patience tout au long de ces années.

Contents

List of Figures

List of Tables

Abstract

During the last 20 years, the joint expansion of computing power, computer graphics, networking capabilities and multiresolution analysis have stimulated several research domains, and developed the need for new types of data such as 3D models, i.e. discrete surfaces. In the intersection between multiresolution analysis and computer graphics, subdivision methods, i.e. iterative refinement procedures of curves or surfaces, have a non-negligible place, since they are a basic component needed to adapt existing multiresolution techniques dedicated to signals and images to more complicated data such as discrete surfaces represented by polygonal meshes. Such representations are of great interest since they make polygonal meshes nearly as flexible as higher level 3D model representations, such as piecewise polynomial based surfaces (e.g. NURBS, B-splines...).

The generalization of subdivision methods from univariate data to polygonal meshes is relatively simple in case of a regular mesh but becomes less straightforward when handling irregularities. Moreover, in the linear univariate case, obtaining a smoother limit curve is achieved by increasing the size of the support of the subdivision scheme, which is not a trivial operation in the case of a surface subdivision scheme without *a priori* assumptions on the mesh. While many linear subdivision methods are available, the studies concerning more general non-linear methods are relatively sparse, whereas such techniques could be used to achieve better results without increasing the size support.

The goal of this study is to propose and to analyze a binary non-linear interpolatory subdivision method. The proposed technique uses local polar coordinates to compute the positions of the newly inserted points. It is shown that the method converges toward continuous limit functions. The proposed univariate scheme is extended to triangular meshes, possibly with boundaries.

In order to evaluate characteristics of the proposed scheme which are not proved analytically, numerical estimates to study convergence, regularity of the limit function and approximation order are studied and validated using known linear schemes of identical support. The convergence criterion is adapted to surface subdivision via a Hausdorff distance-based metric. The evolution of Gaussian and mean curvature of limit surfaces is also studied and compared against theoretical values when available.

An application of surface subdivision to build a multiresolution representation of 3D models is also studied. In particular, the efficiency of such a representation for compression and in terms of rate-distortion of such a representation is shown. An alternate to the initial SPIHT-based encoding, based on the JPEG 2000 image compression standard method. This method makes possible partial decoding of the compressed model in both SNR-progressive and level-progressive ways, while adding only a minimal overhead when compared to SPIHT.

Version abrégée

Durant les 20 dernières années, l'expansion simultanée de la puissance de calcul des ordinateurs, de leurs capacités graphiques et d'interconnection par réseau, ainsi que des techniques liées à l'analyse multirésolution a stimulé de nombreux domaines de recherche et a fait croître les besoins en nouveaux types de données comme par exemple les modèles 3D, i.e. les surface discrètes. Dans la partie commune à l'analyse multirésolution et à la modélisation informatique de courbes et de surfaces, les techniques de subdivision, i.e. les méthodes itératives de raffinement des courbes ou des surfaces, tiennent une place non-négligeable car elles constituent un bloc essentiel à l'adaptation des techniques multirésolution dédiées aux signaux et aux images à des donées plus complquées, comme par exemple les surfaces discrètes représentées par des maillages polygonaux. Ces représentations multirésolutions sont d'un grand intérêt, car elles permettent aux maillages polygonaux d'être presque aussi flexibles que des représentations de plus haut niveau des modèles 3D, comme par exemple les surfaces générées grâces à des polynômes par morceaux (NURBS, B-splines...).

La généralisation des méthodes de subdivision dédiées aux signaux unidimensionnels aux maillages polygonaux est relativement simple dans le cas de maillages réguliers, mais devient plus difficile quand doivent être prises en compte les possibles irrégularités. De plus, alors qu'il est possible d'obtenir des courbes limites plus régulières en augmentant le support d'un schéma de subdivision unidimensionnel linéaire, ceci ne peut que très difficilement être généralisé au cas des méthodes de subdivision pour les surfaces discrètes sans faire de suppositions *a priori* sur le maillage. Alors que beaucoup d'études ont été menées sur les schémas de subdivision linéaires, celles concernant les méthodes non-linéaires sont nettement moins nombreuses, alors que de telles méthodes pourraient être utilisées pour obtenir de meilleurs résultats que les méthodes linéaires de même support.

Le but de cette étude est de proposer et d'analyser une méthode de subdivision non-linéaire, binaire et interpolante pour des signaux unidimensionnels. La méthode proposée utilise les coordonnées polaires locales pour calculer les positions des nouveaux points. Il est démontré que cette méthode génère des fonctions limites continues. Le schéma proposé est également étendu aux maillages triangulaires, ainsi qu'aux maillages triangulaires ayant des bords.

Pour évaluer les autres caractéristiques de la méthode de subdivision proposée qui ne sont pas prouvées analytiquement, des estimateurs numériques pour étudier la convergence, la régularité de la fonction limite et l'ordre d'approximation, sont définis et validés grâce à des schémas linéaires connus ayant le même support que la méthode non-linéaire étudiée. Le critère de convergence est étendu aux méthodes de subdivision pour les surfaces grâce à une métrique basée sur la distance de Hausdorff. L'évolution des courbures Gaussiennes et moyennes des surfaces limites obtenues est également étudiée et comparée aux valeurs théoriques lorsque celles-ci sont disponibles.

Une application de la subdivision des surfaces pour générer des représentations multirésolution de modèles 3D est également étudiée. En particulier, l'intérêt de cette méthode pour compresser les modèles 3D ainsi que son efficacité en termes de débit-distortion est montrée. Une méthode alternative d'encodage, basée sur la méthode utilisée dans le standard de compression d'images JPEG 2000, est proposée pour remplacer la méthode initiale basée sur SPIHT. Cette nouvelle méthode rend possible le décodage partiel, soit progressif par niveau de résolution, soit progressif par qualité, des modèles 3D compressés tout en ne nécessitant que peu d'informations supplémentaires par rapport à SPIHT.

Introduction

1

1.1 Context

Although initially studied in the late 1940s by the swiss mathematician Georges de Rham, subdivision methods had to wait the development of computer graphics, roughly the 1970s, to start being actively studied and improved. Starting from corner-cutting methods that generate smooth curves from a control polygon (this idea of corner-cutting was studied by de Rham and apparently used to build smooth wood hammer handles), these iterative refinement methods have been generalized to be used on more complex types of data.

During the last two decades, the rise of multiresolution analysis gave birth to significant advances in a wide range of domains. Wavelet decomposition of signals or images, which is one of the most obvious and vastly used application of multiresolution analysis, is a valuable tool for building efficient and intuitive hierarchical representations of such data. In addition, the growing interest for efficient algorithms dedicated to 3D models represented by discrete polygonal surfaces, along with the growth of computing power and the increase of network applications make discrete surfaces an attractive field of study.

Recently, several 3D scanning techniques have been developed and made the generation of highly detailed 3D models represented by polygonal meshes, quite easy and inexpensive. The relative complexity and requirements of algorithms dedicated to this type of data make sense for the development of multiresolution-based processing tools. In this multiresolution analysis framework, subdivision algorithms quickly appeared as a key component to adapt multiresolution techniques to discrete surfaces, being often used as the analogous of low-pass reconstruction filter in wavelet construction.

An additional interest of having multiresolution representations of discrete polygonal surfaces appears when compared to the alternate representations used in computer-aided design, i.e. piecewise polynomial representations (based for instance on B-splines or NURBS). Their popularity in the field of geometric modeling is highly correlated with their ease of use to design complex surfaces by changing only a few control points. Once a multiresolution representation of a discrete polygonal mesh is built, its overall shape is entirely controlled by a small number of control points, possibly belonging to different levels in order to achieve coarse or fine control, which finally brings to polygonal meshes a flexibility comparable to that of polynomial-based representations. Since the polygonal mesh representation of a discrete surface is the most generic, and used in graphics hardware (even polynomial-based representations are converted to this format before being rendered), it is crucial to have powerful methods dedicated to this type of data at hand.

1.2 Approach

Linear subdivision, especially in the case of univariate signals, has been widely studied and its properties are well-known. When adapted to discrete surfaces, linear surface subdivision schemes inherit properties from their generating univariate schemes. However, without *a priori* knowledge on the regularity of the discrete surface under consideration, generalizing wide-support univariate schemes is difficult, if not impossible, hence the need to keep the support of a subdivision scheme as small as possible[*].

The approach followed in this study is fundamentally different from most of the literature, since it focuses on a *non-linear* scheme. Even if studies regarding non-linear subdivision can be found, most of them deal with particular cases of non-linear schemes (e.g. "linear" schemes having data-dependent coefficients). The proposed scheme is fairly different from what was addressed so far in these studies. It makes use of a local coordinate system, and takes advantage of the polar (spherical in the case of a surface) coordinates to computes the new midpoint coordinates, which makes its adaptation from univariate data to discrete surfaces straightforward, even in the case of irregular meshes. It is demonstrated that the proposed scheme converges toward continuous functions.

Numerical methods to evaluate the main characteristics of the non-linear scheme are proposed. By applying these numerical tools to both proposed scheme and to known linear schemes of identical support we can formulate conjectures with respect to the behavior of the proposed non-linear scheme.

A particular application of surface subdivision is also proposed, i.e. to build a multiresolution representation of discrete surfaces and to use this representation to compress efficiently the surface. In this method, the subdivision scheme is the analogous of the low-pass reconstruction filter in a lifting scheme. The relation between the subdivision scheme used to build the multiresolution representation and rate-distortion performance of the coding technique are studied. Finally, an alternate encoding method that enables both SNR-progressive and level-progressive partial decoding of the compressed model is proposed and compared to the original SPIHT-based coder.

1.3 Main contributions

The main contribution of this study can be summarized as follows

- Definition of a non-linear univariate interpolatory subdivision scheme, based on local polar coordinates.
- Analysis of the behavior of the scheme and proof of the convergence of the scheme towards continuous functions.
- Adaptation of the scheme to discrete triangular meshes. Generalization of the univariate method to handle mesh boudaries.

---

[*] It is however worth noting that an opposite approach is also possible, i.e. to take *every* control point of the mesh into account to compute each newly inserted point by minimizing an energy functional. This *variational* approach of subdivision will not be developed in this study. We refer the reader to [75] for a presentation of such methods.

- Definition of numerical criteria to study the convergence, regularity of the limit function and approximation order of a univariate subdivision scheme. The convergence criterion is adapted to surface subdivision schemes using a Hausdorff distance-based metric.
- Study of the evolution of Gaussian and mean curvatures of the surface generated through subdivision, and comparison with linear schemes of identical support.
- Evaluation of these numerical criteria by applying them to known linear subdivision schemes, and comparison with the results achieved using the proposed non-linear method.
- Evaluation of the influence of surface subdivision scheme on rate-distortion performance of an existing multiresolution 3D model compression method that uses surface subdivision as predictor.
- Adaptation of an existing bitplane coder used for image compression in the JPEG 2000 standard to replace the SPIHT coder used in the 3D model compression technique, providing both SNR and level-progressive partial decoding capabilities, while adding a minimal overhead to the compressed model size.

1.4 Outline

This dissertation is organized as follows. Chapter 2 will review the existing linear subdivision techniques for univariate data. The analysis tools associated to these methods will also be detailed. Their extensions to less simple cases such as non-stationary and non-uniform subdivision schemes will also be presented. Chapter 3 will describe the generalization of univariate subdivision schemes to discrete surfaces. The different possibilities available to represent discrete surfaces will be briefly discussed. This chapter will review the major linear surface subdivision techniques and their associated analysis tools.

Chapter 4 will be focused on non-linear subdivision, and mainly on the proposed interpolatory non-linear univariate subdivision scheme based on a local coordinate system. The univariate scheme will be analyzed and generalized to a surface subdivision scheme. In Chapter 5, we will introduce several numerical criteria to analyze and evaluate the key parameters of univariate and surface subdivision schemes. We will apply these criteria to compare the proposed scheme with known linear schemes. An application of subdivision is shown in Chapter 6, where surface subdivision schemes are used to build multiresolution representations and compress 3D models. Existing model compression techniques are briefly reviewed before introducing the studied method. Finally, conclusions and possible research directions are presented in Chapter 7.

Linear subdivision of univariate data

2

In this chapter we review classical results concerning linear subdivision of univariate data. The historical aspects of subdivision can be found in most of the references and in [7], which deals mostly with Bézier and spline curves/surfaces. Alternative presentations of similar concepts can be found in [136], [63] (in the chapters written by N. Dyn and M. Sabin), [35] and [45].

This chapter is organized as follows: Section 2.1 will present the fundamental notations, definitions regarding the representation and fundamental properties of subdivision schemes. Some well known examples of subdivision schemes will be presented in Section 2.2. The analysis tools for stationary schemes will be detailed in Section 2.3, and extended to non-stationary and non-uniform schemes in Section 2.4.

2.1 Notation - Definitions

This section presents several core concepts and notations that will be used throughout the chapter, such as multi-level grids, subdivision operators and their convergence.

2.1.1 Univariate data and multi-level grids

As stated in the title of the chapter, the focus will be on discrete *univariate data*, i.e. a countable number of samples from a finite-dimension convex part of a vector space. In this study, we will assume that this data is represented by a mapping between $\mathbb{Z}$ (or a subset) and $\mathbb{R}^d$, with $d \geq 1$, which implicitely defines a polygonal line joining all those data points.

As will be seen later, subdivision introduces the concept of *refinement* of such data, and the underlying concept of *level of resolution.* In order to deal with discrete univariate data with several level of resolution, we will use *multi-level grids.* Let us denote by $X_j, j \in \mathbb{N}$ the grid at level j, which is a set of real numbers

$$X_j = \{x_k^j \in \mathbb{R} \mid k \in \mathbb{Z}\}.$$

A grid X_j is *strictly increasing* if $x_k^j < x_{k+1}^j$, $\forall k \in \mathbb{Z}$. A sequence of grids $\{X_j\}_{j\in\mathbb{N}}$ is *nested* if for all j, the property $X_j \subset X_{j+1}$ holds. The distances between any two elements of the grid X_j are denoted by

$$\begin{aligned} d_k^{j\,[p]} &= x_{k+p}^j - x_k^j,\ p \in \mathbb{N}, \text{ and} \\ d_k^j &= d_k^{j\,[1]} = x_{k+1}^j - x_k^j. \end{aligned}$$

A grid X_j is called *uniform* if all the $d_k^j = d_j\ \forall k$. A particular case of nested sequences of grids $\{X_j\}$ is the uniform one generated by *dyadic refinement*, i.e. when X_0 is uniform and the following property holds,

$$d_{2k}^{j+1} = d_{2k+1}^{j+1} = \frac{d_k^j}{2},\ \forall (j,k) \in \mathbb{N} \times \mathbb{Z}.$$

In such a grid, the distance between two consecutive points shrinks by a factor 2 each time j increases. Moreover, such a grid is equivalent to the uniform dyadic grid defined by $x_k^j = k2^{-j}$.

Definition 1. *Let $j \in \mathbb{N}$. Let us consider a grid $X_j = \{x_k^j\}_{k\in\mathbb{Z}}$ and a set $F_j = \{f_k^j \in \mathbb{R}^d\}_{k\in\mathbb{Z}}$, where d in a non-zero positive integer. The couple (X_j, F_j) defines a discrete univariate function f_j at the level of resolution j, which associates to each element of X_j a real number from F_j*

$$\begin{aligned} f_j : X_j &\longrightarrow F_j \\ x_k^j &\longmapsto f_k^j. \end{aligned}$$

By extension, we will always consider f_j as the piecewise linear function that satisfies the above mapping.

For instance, a polygon from $\mathbb{R}^2$ having n vertices can be represented by the grid (X, F) with $X = \{1, 2, \ldots, n\}$ and $F = \{v_k \in \mathbb{R}^2,\ k \in X\}$ where the v_k denote the coordinates of the vertices forming the polygon.

We will often assume (especially when $j = 0$, i.e. at the coarsest level) that the elements from f_j are samples from a $\mathcal{C}^n$ function f (with $n > 0$), i.e. $f_k^j = f(x_k^j)\ \forall k$.

In order to study the regularity of such discrete functions, especially on non-uniform grids, *finite differences* are used. The first order forward finite difference of f_j at k, denoted by $f_k^{j\,[1]}$, is defined by

$$f_k^{j\,[1]} = \frac{f_{k+1}^j - f_k^j}{d_k^j}.$$

Finite differences of order $p > 1$ at k are defined recursively using the relation

$$f_k^{j\,[p]} = \frac{f_{k+1}^{j\ \,[p-1]} - f_k^{j\,[p-1]}}{d_k^{j\,[p]}}.$$

2.1.2 Subdivision operators

The core concept on which rely this study is the notion of *subdivision operator*. The effect of this operator on a discrete function, represented by (X_j, F_j), is to generate the next level, i.e. (X_{j+1}, F_{j+1}). This subdivision step can be expressed using a *refinement equation*

$$(X_{j+1}, F_{j+1}) = \mathcal{S}_j(X_j, F_j), \tag{2.1}$$

where $\mathcal{S}_j$ denotes the subdivision operator at level j. The set of subdivision operators $\{\mathcal{S}_j\}$ will be denoted by $\mathcal{S}$. While this is the most generic way of defining subdivision, most of the subdivision operators usually have additional constraints, which are detailed hereafter.

Definition 2. *The subdivision is* ***interpolating*** *if all the data points belonging to the function at level j also belong to the function at level $j+1$. Using the notation conventions previously defined, let us consider a data point of index k and level j, $(x_k^j, f_k^j) \in X_j \times F_j$. In case of an interpolating subdivision operator, there exists $k' \in \mathbb{Z}$ such that*

$$(x_k^j, f_k^j) = (x_{k'}^{j+1}, f_{k'}^{j+1}),\ \forall (j,k) \in \mathbb{N} \times \mathbb{Z}.$$

This also implies that $X_{j+1} \subset X_j$ and $F_{j+1} \subset F_j$.

In the case of subdivision operators verifying this condition, the sequence of grids/functions generated by applying this process several times are obviously nested. If the subdivision operator is not interpolating, it is *approximating.*

Interpolating subdivision sticks to the idea of *refining* a grid, allowing to generate different levels of resolution starting from an initial coarse grid, or control polygon if dealing with plane curves. Definition 2 does not forbid to have a number of new points inserted between two consecutive points from level j that varies with k.[*] However, we will generally assume to have this number to remain constant, i.e. to have $n-1$ new points inserted between two consecutive samples from level j. This restriction of Definition 2 is referred to as "n-ary" subdivision ($n \geq 2$). The quantity n is usually referred to as the *arity* of a scheme. In the case of non-interpolating subdivision, the arity is the number of "new" points that is associated to each "old" point. For instance, the main focus of this study will be *binary* interpolating subdivision, which leads to the following relations,

$$\begin{aligned} x_{2k}^{j+1} &= x_k^j, \\ f_{2k}^{j+1} &= f_k^j, \end{aligned} \tag{2.2}$$

and the new points lie at $(x_{2k+1}^{j+1}, f_{2k+1}^{j+1})$.

A special case of subdivision, but widely used, is *linear* subdivision.

Definition 3. *A subdivision operator is* ***linear*** *if it consists of linear combination of data points from level j to generate level $j+1$, i.e. for all j and k, there exists two sets of real numbers (often called* ***subdivision masks*** *or* ***stencils****), $\{\alpha_m^j\}_{(j,m)\in\mathbb{N}\times\mathbb{Z}}$ and $\{\beta_m^j\}_{(j,m)\in\mathbb{N}\times\mathbb{Z}}$ such that*

$$\begin{aligned} f_k^{j+1} &= \sum_l \alpha_{k-nl}^j f_l^j, \\ x_k^{j+1} &= \sum_l \beta_{k-nl}^j x_l^j. \end{aligned} \tag{2.3}$$

In the above relation, n is the "arity" of the scheme, e.g. in the case of a binary scheme, we have $n=2$. It is important to note that in the above relations, the coefficients α_{k-nl}^j and β_{k-nl}^j could depend on x and f. In such a case, the operator is ***data-dependent****. When the sums involve a finite number of samples, the subdivision operator is said to have a* ***finite support****. If the α and β coefficients do not depend on j, the scheme is* ***stationary****. Similarly, if the coefficients do not depend on k, the scheme is termed* ***uniform****. In general, referring to non-uniform subdivision includes non-stationary subdivision.*

In the case of binary, finite-support, stationary and uniform refinement rules, on a uniform grid, equation (2.3) becomes

$$\begin{aligned} x_m^{j+1} &= m2^{-(j+1)}, \\ f_m^{j+1} &= \sum_{l=0}^{N} \alpha_{m-2l} f_l^j. \end{aligned} \tag{2.4}$$

---

[*] In fact, this can be useful when trying to *adapt* the subdivision to the grid, for instance by inserting more points where the data show high variations.

2.1.3 Convergence

In order to define the limits of a discrete function lying on a grid, the domain of definition (i.e. the $\{x_j\}$) of the function must have a limit that is dense in $\mathbb{R}$ (or in a compact set of $\mathbb{R}$). In the following, we will assume that such a property is verified, which is obviously true when dealing with using the canonical uniform grid $x_k^j = k2^{-j}$. Let us denote by K the limit compact of the $\{x_j\}$ sequence.

Let (X_0, F_0) be an initial coarse grid, and $\mathcal{S}$ a subdivision operator. The sequence of piecewise linear functions $(f_j)_{j\in\mathbb{N}}$, obtained by successive subdivisions of f_0 using the refinement equation (2.1), *converges* on K if there exists a function f such that for all $x \in K$

$$\lim_{j\to\infty} |f_j(x) - f(x)| = 0.$$

This definition of convergence is unfortunately too basic, since it only guarantees pointwise convergence. In order to have more interesting properties concerning the limit function, *uniform convergence* is required.

Definition 4. *The sequence of piecewise linear functions f^j **converges uniformly** on K if there exists a function f such that*

$$\lim_{j\to\infty} \sup_{x\in K} |f^j(x) - f(x)| = 0.$$

The following sufficient condition is useful to prove uniform convergence.

Theorem 1. *If there exists $0 < \alpha < 1$ and $\beta > 0$ such that for all $j \in \mathbb{N}$*

$$\|f_j - f_{j-1}\|_\infty \leq \beta\alpha^{j-1},$$

the sequence $(f^j)_{j\in\mathbb{N}}$ converges uniformly toward a limit function.

Uniform convergence of sequences of functions provides valuable information concerning the regularity of the limit function, as detailed in the following theorem.

Theorem 2. *If f exists, it is continuous, since all the f_j are continuous by construction and the sequence $(f^j)_{j\in\mathbb{N}}$ converges uniformly.*

Proof. Let $(x, x_0) \in K^2$. The quantity $f(x) - f(x_0)$ can be written as

$$f(x) - f(x_0) = (f(x) - f^j(x)) + (f^j(x) - f^j(x_0)) + (f^j(x_0) - f(x_0)),$$

which yields

$$|f(x) - f(x_0)| \leq |f(x) - f^j(x)| + |f^j(x) - f^j(x_0)| + |f^j(x_0) - f(x_0)|. \tag{2.5}$$

The sequence (f^j) converges uniformly to f. Therefore, for any $\varepsilon > 0$, there exists $p \in \mathbb{N}$ such that for all $j > p$, and for all x, we have $|f^j(x) - f(x)| < \varepsilon$. Using this property in the inequality (2.5) leads to

$$|f(x) - f(x_0)| < 2\varepsilon + |f^j(x) - f^j(x_0)|$$

Using the fact that f^j is continuous, for all $\varepsilon > 0$, there exists $\eta > 0$ such that $|x - x_0| < \eta$ implies $|f^j(x) - f^j(x_0)| < \varepsilon$. Using the previous inequality, $|x - x_0| < \eta$ implies $|f(x) - f(x_0)| < \varepsilon' = 3\varepsilon$, which proves continuity. □

The goal of subdivision being to obtain as smooth as possible limit functions, the above definitions and theorems need to be extended.

Definition 5. *A uniformly convergent subdivision scheme is called* $\mathcal{C}^p$ *if for any initial grid, the limit function has continuous derivatives up to order* p.

While the smoothness of the limit function is primordial, another criterion is worth being considered, namely the *approximation order*. The underlying concept is intuitive: if we consider an initial grid obtained by sampling a sufficiently regular function, the error between the original function and the limit function obtained through subdivision should decrease along with the sampling step.

Definition 6. *Let us consider the initial grid* $X_0 = \delta\mathbb{Z}$, $\delta \in \mathbb{R}$ *and initial data* $f_{0,k} = g(k\delta)$ *sampled from a function* $g \in \mathcal{C}^k$ *defined over* $\mathbb{R}$. *Let us denote by* f_∞ *the limit function obtained. The subdivision scheme has approximation order* p *if*

$$|(g - f_\infty)(x)| \leq C\delta^p, \forall x \in \mathbb{R},$$

where C *is a real constant.*

Most of the finite-support linear and stationary schemes are $\mathcal{C}^1$ or $\mathcal{C}^{2-\varepsilon}$, i.e. $\mathcal{C}^2$ everywhere except over a countable set of points where they are only $\mathcal{C}^1$. We refer to Section 2.3 for an in-depth analysis of the topic.

2.2 Examples of univariate subdivision schemes

In this section, some examples of linear schemes are presented. They will also be used in Section 2.3 when giving the details of the regularity analysis.

2.2.1 "Corner cutting" methods

The earliest subdivision schemes proposed were based on the idea of "corner cutting". This idea was first formulated in 1947, by Georges de Rham [24, 25], and used a simple "trisection rule" to refine polygons: *the points dividing each side of a polygon into three parts of equal length are the vertices of the next polygon.* As explained by de Rham in his paper, such a technique was used to create cylindrical hammer handles. This rule is illustrated in Figure 2.1. The initial polygonal line ABC becomes $A'B_1B_2C'$ in the next level. The trisection rule yields the following relations

$$\|AA'\| = \|A'B_1\| = \|B_1B\| = \frac{1}{3}\|AB\|,$$

and

$$\|BB_2\| = \|B_2C'\| = \|C'C\| = \frac{1}{3}\|BC\|.$$

Let us write the trisection rule using the notation conventions and definitions from the previous sections. Let us consider a sequence of polygonal lines, defined by the $(X_j, F_j)_{j\in\mathbb{N}}$. In case of a closed polygon, the initial grid will have a finite number of points. The refinement rules to get the level $j+1$ from level j is

$$\begin{aligned}
\begin{pmatrix} x_{2k}^{j+1} \\ f_{2k}^{j+1} \end{pmatrix} &= \frac{2}{3}\begin{pmatrix} x_k^j \\ f_k^j \end{pmatrix} + \frac{1}{3}\begin{pmatrix} x_{k+1}^j \\ f_{k+1}^j \end{pmatrix}, \\
\begin{pmatrix} x_{2k+1}^{j+1} \\ f_{2k+1}^{j+1} \end{pmatrix} &= \frac{1}{3}\begin{pmatrix} x_k^j \\ f_k^j \end{pmatrix} + \frac{2}{3}\begin{pmatrix} x_{k+1}^j \\ f_{k+1}^j \end{pmatrix}.
\end{aligned}$$

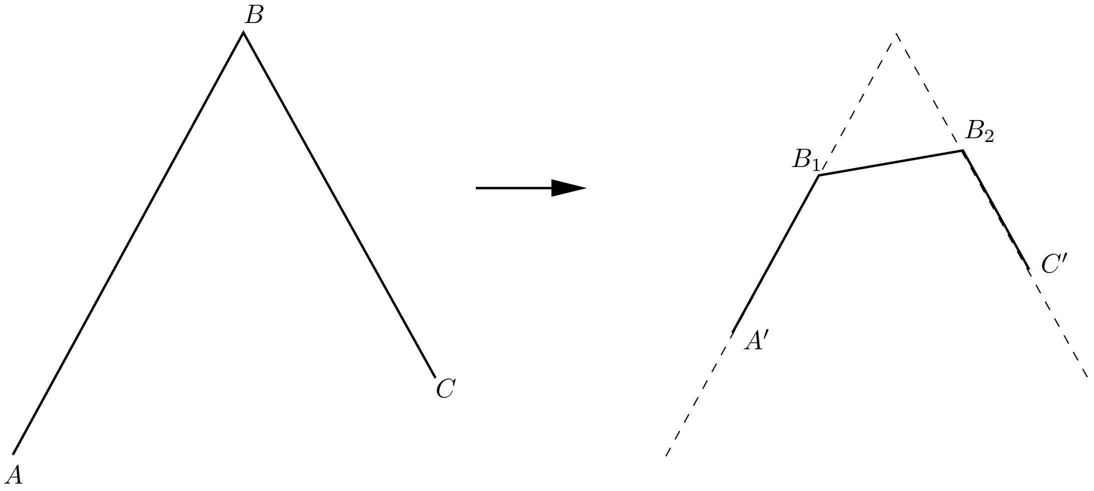


Figure 2.1: Cutting a corner using the trisection rule.

In a further study concerning the limit curves obtained by trisection [26], de Rham considered a more general rule,

$$\begin{aligned} \begin{pmatrix} x_{2k}^{j+1} \\ f_{2k}^{j+1} \end{pmatrix} &= (1-\beta_1)\begin{pmatrix} x_k^j \\ f_k^j \end{pmatrix} + \beta_1 \begin{pmatrix} x_{k+1}^j \\ f_{k+1}^j \end{pmatrix}, \\ \begin{pmatrix} x_{2k+1}^{j+1} \\ f_{2k+1}^{j+1} \end{pmatrix} &= \beta_2 \begin{pmatrix} x_k^j \\ f_k^j \end{pmatrix} + (1-\beta_2)\begin{pmatrix} x_{k+1}^j \\ f_{k+1}^j \end{pmatrix}, \end{aligned} \tag{2.6}$$

where β_1 and β_2 are two positive real numbers such that $0 < \beta_1 + \beta_2 < 1$.

In 1974, G. Chaikin used a similar rule to generate what turned out to be quadratic B-splines [112] from a polygonal line, by using a recursive algorithm with a restricted set of operations. Chaikin's corner cutting rule [11] follows the method described in equations (2.6), with $\beta_1 = \beta_2 = \frac{1}{4}$. Figures 2.2 and 2.3 show respectively the effect of Chaikin's rule and de Rham's trisection ($\beta_1 = \beta_2 = \frac{1}{3}$) on a closed polygon. Using the classification criteria of Section 2.1.2, these "corner cutting" refinement rules are linear, finite-support, and stationary subdivision operators. Moreover, they are not interpolating, but approximating.

The mask, as defined in (2.4), for Chaikin's rule is given in Table 2.1.

k		-2	-1	0	1	
a_k	$\dots 0$	$\frac{1}{4}$	$\frac{3}{4}$	$\frac{3}{4}$	$\frac{1}{4}$	$0\dots$

Table 2.1: Chaikin's rule mask

It is interesting to note that higher order version of Chaikin's rule can be derived from the "Box-Splines" subdivision techniques. We refer to M. Sabin's chapter in [63] for an intuitive construction of such schemes, and to [23] for formal definitions of box splines. It is interesting to remark that all the coefficients for the scheme of order p can be derived from the $p+1$'th row of Pascal's triangle. The mask of the scheme of order p is given by

$$a_{p,k} = 2^{-p}\binom{p+1}{k}. \tag{2.7}$$

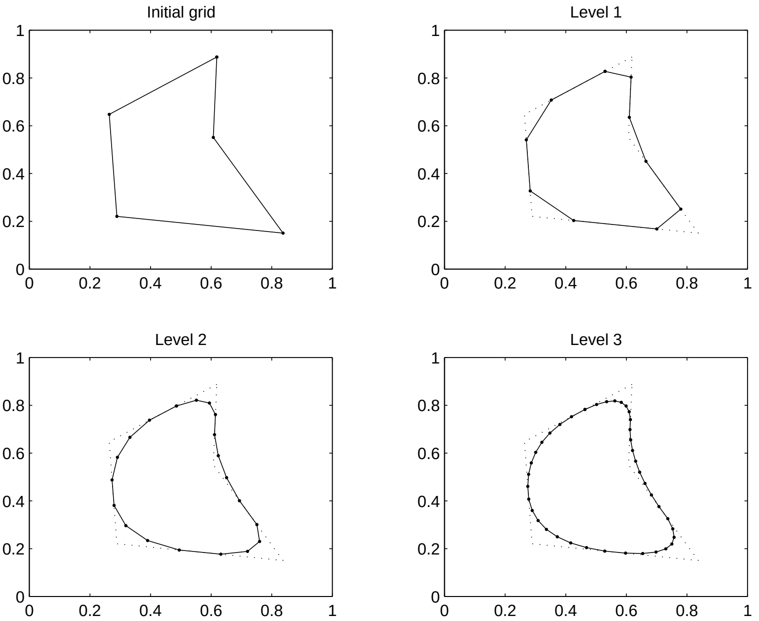


Figure 2.2: Chaikin's rule over a closed polygon, over 3 levels.

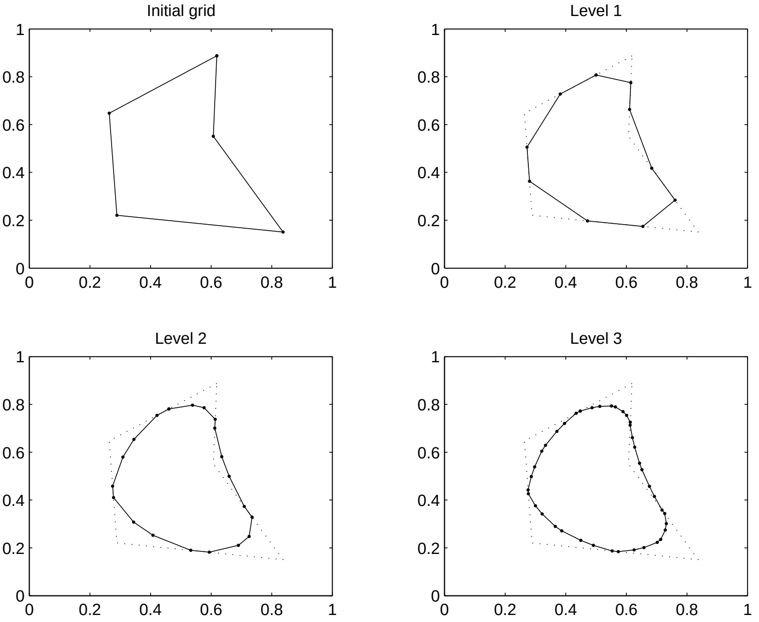


Figure 2.3: de Rham's trisection rule over a closed polygon, over 3 levels.

As an example, the cubic approximation rule is defined by

$$\begin{aligned} \begin{pmatrix} x_{2k}^{j+1} \\ f_{2k}^{j+1} \end{pmatrix} &= \frac{4}{8}\begin{pmatrix} x_k^j \\ f_k^j \end{pmatrix} + \frac{4}{8}\begin{pmatrix} x_{k+1}^j \\ f_{k+1}^j \end{pmatrix}, \\ \begin{pmatrix} x_{2k+1}^{j+1} \\ f_{2k+1}^{j+1} \end{pmatrix} &= \frac{1}{8}\begin{pmatrix} x_k^j \\ f_k^j \end{pmatrix} + \frac{6}{8}\begin{pmatrix} x_{k+1}^j \\ f_{k+1}^j \end{pmatrix} + \frac{1}{8}\begin{pmatrix} x_{k+2}^j \\ f_{k+2}^j \end{pmatrix}. \end{aligned}$$

A comparison between the limit curves obtained using those box-splines rules of degree 2 to 5 is shown in Figure 2.4

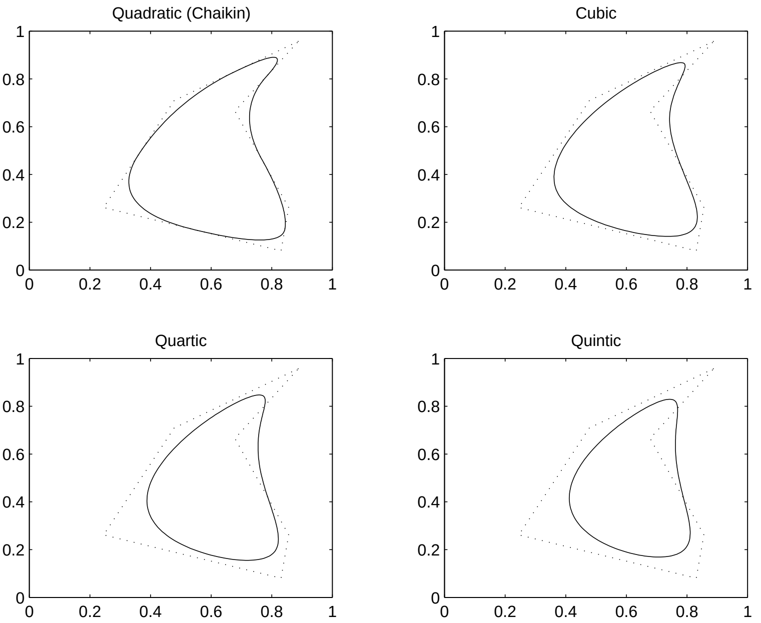


Figure 2.4: Box-spline rules of degree 2 to 5, after 5 iterations (the control polygon is shown by the dotted line)

2.2.2 Deslauriers-Dubuc interpolating schemes - Four-point scheme

Although the approximating methods described in the previous paragraph give interesting results, it is often required for particular applications to have interpolating schemes. The first works regarding linear interpolating schemes were done by Serge Dubuc in [30, 34]. This interpolation technique, which is simple and robust has been extended and studied in many ways, since it has been used as starting point for many other subdivision techniques.

This method deals with dyadic uniform grids. Let us consider an initial grid (X_0, F_0) with $X_0 = \mathbb{Z}$. The principle is to assimilate the initial values as samples from a cubic polynomial p, i.e. $f_k^0 = p(k)$. At the following level, such an assertion leads to $f_{2k+1}^1 = p\left(k + \frac{1}{2}\right)$. The approach used by Dubuc was to use the cubic Lagrange interpolation polynomial for p, which in the case of a regular grid has constant coefficients. Computing $p\left(k + \frac{1}{2}\right)$ leads to the following interpolation rule, called *four-point scheme*,

$$f_{2k+1}^{j+1} = -\frac{1}{16} f_{k-1}^j + \frac{9}{16} f_k^j + \frac{9}{16} f_{k+1}^j - \frac{1}{16} f_{k+2}^j. \tag{2.8}$$

Almost simultaneously, but independently, a group of subdivision methods was proposed by by Nira Dyn, John A. Gregory and David Levin, briefly in [41], and in a more detailed way in [36]. The idea was to use the following interpolation rule

$$f_{2k+1}^{j+1} = -w f_{k-1}^j + \left(\frac{1}{2} + w\right) f_k^j + \left(\frac{1}{2} + w\right) f_{k+1}^j - w f_{k+2}^j, \text{ with } w > 0. \tag{2.9}$$

In this definition, w stands for a *tension parameter.* Although following different paths, the results regarding the four-point scheme converged, since equation (2.8) is obtained for $w = \frac{1}{16}$, which is a value for which the refinement equation (2.9) produces a smooth limit curve (see Section 2.3). Figure 2.5 shows the effect of the four-point scheme over the same initial polygon used to demonstrate the effects of corner-cutting rules.

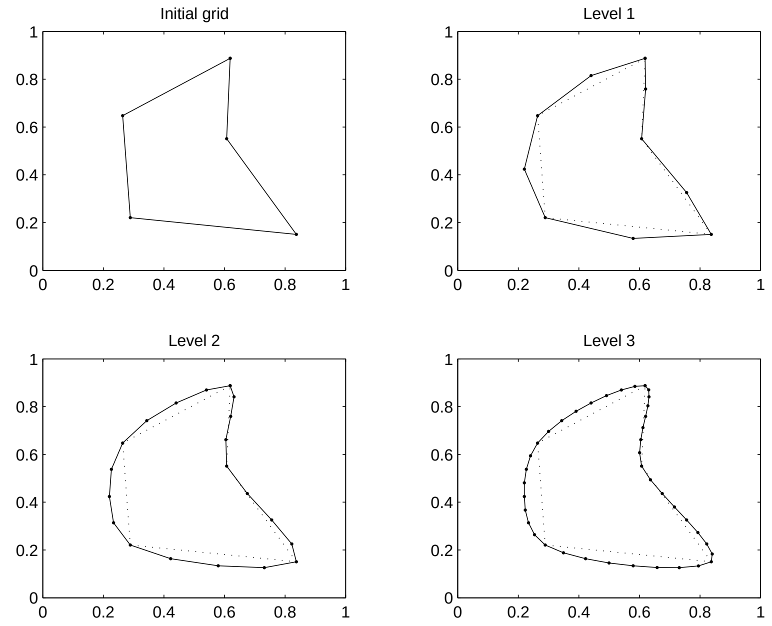


Figure 2.5: Four-point interpolating subdivision over a closed polygon, over 3 levels.

The mask of the four-point scheme, as defined in (2.4) is given in Table 2.2. The constraint $a_0 = 1$ is implied by the fact that the scheme is interpolatory.

k		-3	-2	-1	0	1	2	3	
a_k	$\ldots 0$	$-w$	0	$\frac{1}{2}+w$	1	$\frac{1}{2}+w$	0	$-w$	$0\ldots$

Table 2.2: Binary four-point scheme's mask

This technique of interpolation using Lagrange interpolation polynomials has been generalized, by using higher degree polynomials [31]. However, the four-point scheme remained the "reference" in the field of interpolating subdivision, because it produced quite interesting results (cf. Figure 2.5), providing a nearly-optimal trade-off between a reduced complexity and an acceptable smoothness of the limit curve, with respect to higher degree schemes. As developed in Chapter 3, it has also been extended to discrete surfaces, for which a reduced support is a significant advantage.

2.2.3 Four-point ternary scheme

Although they form the most common class of subdivision schemes, binary schemes are not the only ones. An interesting example of ternary stationary interpolating scheme can be found in [59]. It is termed "four-point" scheme because of its support, being identical to the binary four-point scheme described above. The considered ternary scheme uses the following interpolation rule

$$\begin{aligned}
f_{3k}^{j+1} &= f_k^j,\\
f_{3k+1}^{j+1} &= \alpha_0 f_{k-1}^j + \alpha_1 f_k^j + \alpha_2 f_{k+1}^j + \alpha_3 f_{k+2}^j,\\
f_{3k+2}^{j+1} &= \alpha_3 f_{k-1}^j + \alpha_2 f_k^j + \alpha_1 f_{k+1}^j + \alpha_0 f_{k+2}^j,
\end{aligned}$$

where the coefficients a_i are

$$\begin{aligned}
\alpha_0 &= -\frac{1}{18} - \frac{\mu}{6},\\
\alpha_1 &= \frac{13}{18} + \frac{\mu}{2},\\
\alpha_2 &= \frac{7}{18} - \frac{\mu}{2},\\
\alpha_3 &= -\frac{1}{18} + \frac{\mu}{6},
\end{aligned}$$

and $\frac{1}{15} < \mu < \frac{1}{9}$. Figure 2.6 shows the effect of such a rule on a closed polygon, with $\mu = \frac{1}{11}$.

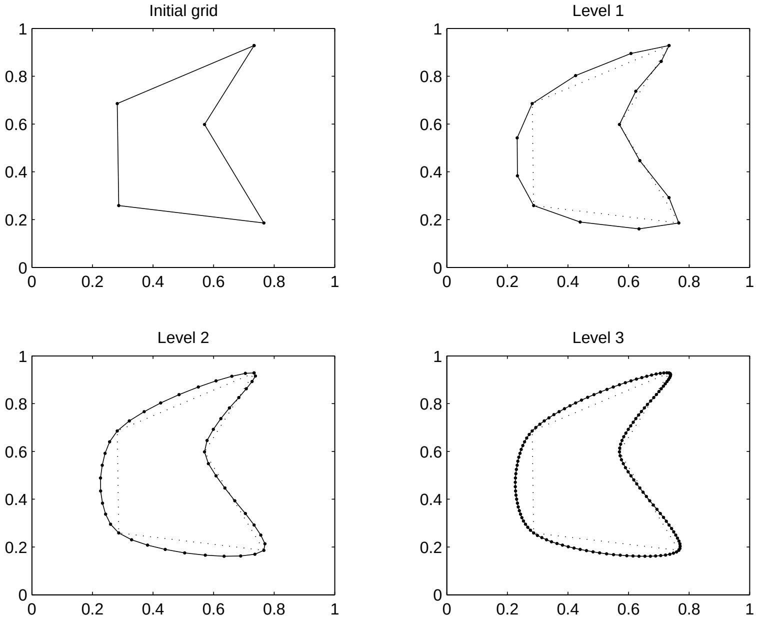


Figure 2.6: Ternary four-point scheme applied to a closed polygon ($\mu = \frac{1}{11}$).

The mask of such a scheme, shown in Table 2.3, is built using the principles of equation (2.3), with $n = 3$ in the refinement equation.

k		-5	-4	-3	-2	-1	0	1	2	3	4	5	
a_k	$\ldots 0$	α_3	α_0	0	α_2	α_1	1	α_1	α_2	0	α_0	α_3	$0\ldots$

Table 2.3: Ternary four-point scheme's mask

2.2.4 Adaptive binary four-point scheme

A more generic version of the four point scheme can be derived when performing subdivision over irregular grids. For instance, starting from a grid $X_0 = \{x_k^0\}$, and deriving the next grids using the relation proposed in [17],

$$\begin{aligned} x_{2k+1}^{j+1} &= \beta x_k^j + (1-\beta)x_{k+1}^j, \\ x_{2k+3}^{j+1} &= (1-\beta)x_{k+1}^j + \beta x_{k+2}^j, \end{aligned}$$

where $0 < \beta < 1$. Using the four-point scheme directly to subdivide data lying on such a grid does not make any sense, since we are no longer trying to compute a value interpolating the midpoint of the interval $[x_k^j, x_{k+1}^j]$. An example of the results obtained are shown in Figure 2.7.

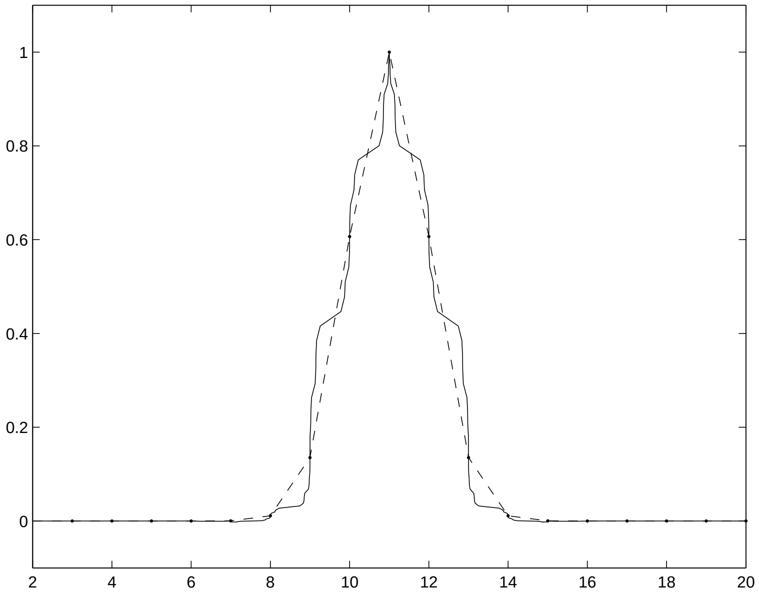


Figure 2.7: "Regular" four-point scheme applied to an irregular grid ($\beta = 0.15$, 4 levels). The dashed line is the initial (regular) grid, sampled from a Gaussian.

In order to be able to handle such cases, it is necessary to come back to the original definition of the four-point scheme, i.e. derive the Lagrange cubic polynomial p that interpolates the values $\{f_{k-1}^j, f_k^j, f_{k+1}^j, f_{k+2}^j\}$ and compute its value at x_{2k+1}^{j+1}. The result can be expressed as follows [17]

$$f_{2k+1}^{j+1} = \sum_{l=k-1}^{k+1} \alpha_{k,l}^j f_l^j,$$

with

$$
\begin{aligned}
\alpha_{k,k-1}^{j} &= -\frac{d_{2k}^{j+1^{[1]}}\, d_{2k+1}^{j+1^{[1]}}\, d_{2k+1}^{j+1^{[3]}}}{d_{2k-2}^{j+1^{[2]}}\, d_{2k-2}^{j+1^{[4]}}\, d_{2k-2}^{j+1^{[6]}}},\\
\alpha_{k,k}^{j} &= \frac{d_{2k-2}^{j+1^{[3]}}\, d_{2k+1}^{j+1^{[1]}}\, d_{2k+1}^{j+1^{[3]}}}{d_{2k-2}^{j+1^{[2]}}\, d_{2k}^{j+1^{[2]}}\, d_{2k}^{j+1^{[4]}}},\\
\alpha_{k,k+1}^{j} &= \frac{d_{2k-2}^{j+1^{[3]}}\, d_{2k}^{j+1^{[1]}}\, d_{2k+1}^{j+1^{[3]}}}{d_{2k-2}^{j+1^{[4]}}\, d_{2k}^{j+1^{[2]}}\, d_{2k+2}^{j+1^{[2]}}},\\
\alpha_{k,k+2}^{j} &= -\frac{d_{2k-2}^{j+1^{[3]}}\, d_{2k}^{j+1^{[1]}}\, d_{2k+1}^{j+1^{[1]}}}{d_{2k-2}^{j+1^{[6]}}\, d_{2k}^{j+1^{[4]}}\, d_{2k+2}^{j+1^{[2]}}}.
\end{aligned}
\tag{2.10}
$$

In the case of an equally-spaced grid, these coefficients become the ones of the four-point scheme of equation (2.8). Figure 2.8 presents the results obtained on the same irregular grid used for Figure 2.7.

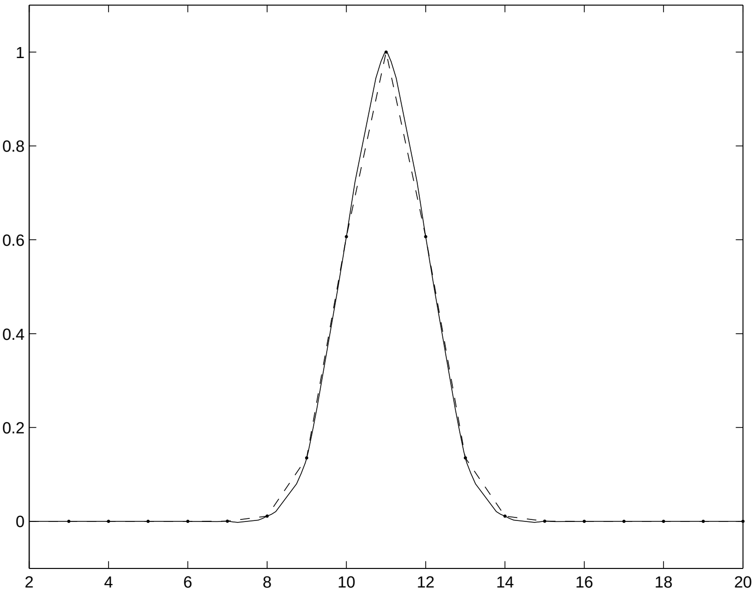


Figure 2.8: "Adaptive" four-point scheme applied to an irregular grid ($\beta = 0.15$, 4 levels).

2.3 Analysis of stationary subdivision schemes

Most of the original papers introducing new subdivision schemes gave a specific proof of the convergence of the proposed method. A proof of the convergence of any corner-cutting method has been done in [22] by Carl de Boor, and the conditions for $\mathcal{C}^1$ continuity of the limit curve have been derived in [55]. Some generic studies of subdivision schemes, not limited to corner-cutting rules have been performed, starting with stationary and uniform subdivision. One way to study subdivision techniques was to consider refinement equations, which are similar to the two-scale equations used in the domain of wavelets and multiresolution analysis. The key studies of these equations applied to curve refinement were done in [96], and developed in [19, 20, 28]. An important study has been done in [10] by Cavaretta, Dahmen and Micchelli, introducing the idea of studying a scheme using a Laurent polynomial associated to the subdivision scheme. A complementary idea, used for the four-point scheme in the initial paper [41] and in [37], was to use matrix formalism. The eigenvalues of matrices representing subdivision schemes can be used to derive the properties of a scheme with respect to the convergence and regularity of the limit function [35]. For the sake of completeness, let us also mention that Kobbelt in [73] chose another approach to study the convergence of interpolatory schemes, considering them as low-pass filters and using their Fourier transform. A good summary of the results concerning uniform stationary binary subdivision can be found in [63]. In this section, we sum up several results regarding the convergence and regularity of *stationary* linear univariate schemes. The relation between regularity and the basic limit function will be detailed in Section 2.3.1, showing the connection between subdivision and multiresolution analysis. The main analysis tool, i.e. Laurent polynomials will be presented in Section 2.3.2, and applied to two classical schemes in Section 2.3.3. The concepts of eigenanalysis will also be briefly introduced in Section 2.3.4.

2.3.1 Basic limit functions and multiresolution analysis

Under the assumption of dealing with uniform binary linear subdivision schemes, a natural approach is to consider the *basic limit function.* This function is defined as follows: let us consider the unit impulsion δ as initial grid, i.e. $X_0 = \mathbb{Z}$ and $F_0 = \{f_k^0\}_{k\in\mathbb{Z}}$ with

$$f_k^0 = \begin{cases} 1 & \text{if } k = 0 \\ 0 & \text{otherwise} \end{cases}.$$

Let us denote by $\mathcal{S}$ a linear scheme that converges and by $\phi_{\mathcal{S}}$ the limit function, i.e. $\phi_{\mathcal{S}} = \mathcal{S}\delta$. This basic limit function can be used to express any limit function generated from the initial data $\{f_k^0\}$ lying on $\mathbb{Z}$

$$\mathcal{S}f_0(x) = \sum_{k\in\mathbb{Z}} f_k^0 \phi_{\mathcal{S}}(x-k). \tag{2.11}$$

It is clear from equation (2.11) that the regularity of the limit function will directly depend on the regularity of the basic limit function. If the scheme is interpolatory, the basic limit functions moreover verifies $\phi_{\mathcal{S}}(k) = \delta_0(k)$. When the scheme is stationary (with coefficients a_k), $\phi_{\mathcal{S}}$ satisfies the refinement equation

$$\phi_{\mathcal{S}}(x) = \sum_{k\in\mathbb{Z}} a_k \phi_{\mathcal{S}}(2x-k). \tag{2.12}$$

Proof. Let us denote by S the operator performing one level of subdivision. If we evaluate $(S\delta_0)$ at $\frac{k}{2}$ using equation (2.4), we have

$$(S\delta_0)\left(\frac{k}{2}\right) = a_k,$$

where the a_k are the subdivision coefficients as defined in equation (2.4). Equation (2.11) rewritten to take into account a grid not being $\mathbb{Z}$ but $h\mathbb{Z}$ (where h is a sampling step) yields

$$\mathcal{S}f_0(x) = \sum_{k\in\mathbb{Z}} f_k^0 \phi_{\mathcal{S}}\left(\frac{x}{h} - k\right).$$

Thus, applying this to $(S\delta_0)$, i.e. $h = \frac{1}{2}$ gives

$$\mathcal{S}(S\delta_0)(x) = \phi_{\mathcal{S}}(x) = \sum_k (S\delta_0)\left(\frac{k}{2}\right)\phi_{\mathcal{S}}(2x-k),$$

which proves (2.12). □

The study of such refinement equations, for instance the one from Daubechies and Lagarias in [19, 20], was of interest to subdivision techniques and also to multiresolution analysis. Multiresolution spaces, as defined by Mallat [92], are used to compute an approximation of a function $f \in L^2_{\mathbb{R}}$ at a resolution 2^{-j}. The approximation of a function at a resolution 2^{-j} is the orthogonal projection of this function on a closed subspace V_j of $L^2_{\mathbb{R}}$. If $f_j \in V_j$ is the approximation of f, it minimizes $\|f - f_j\|$. The subspaces V_j are analogous to the "grids" defined in Section 2.1.1, except that j is not restricted to $\mathbb{N}$, but is considered to belong to $\mathbb{Z}$. This extension is natural, by considering the *subsampling* to go from level j to level $j-1$. Thus, the level -1 will be obtained by subsampling the grid of level 0 by a factor n. In the case of multiresolution analysis (and of binary subdivision), we have $n = 2$.

Definition 7. *The sequence of spaces $\{V_j\}$ defines a* ***multiresolution approximation*** *of functions from $L^2_{\mathbb{R}}$ if it meets the following requirements :*

$$\begin{aligned}
\forall (j,k) \in \mathbb{Z}^2, \quad f(t) \in V_j &\iff f(t - 2^{-j}k) \in V_j, && (2.13)\\
\forall j \in \mathbb{Z}, \quad & \quad\; V_j \subset V_{j+1}, && (2.14)\\
\forall j \in \mathbb{Z},\ \forall t, \quad f(t) \in V_j &\iff f(2^{-1}t) \in V_{j-1}, && (2.15)\\
\lim_{j \to -\infty} V_j = & \quad \bigcap_{j \in \mathbb{Z}} V_j = \{0\}, && (2.16)\\
\lim_{j \to +\infty} V_j = & \quad \overline{\left(\bigcup_{j \in \mathbb{Z}} V_j \right)} = L^2_{\mathbb{R}}, && (2.17)
\end{aligned}$$

and there exists a function θ such that $\{\theta(t-n)\}_{n \in \mathbb{Z}}$ is a Riesz basis of V_0.

The condition in (2.13) states that V_j is invariant with respect to translations. Equation (2.14) is similar to the assumption of having nested grids (e.g. for interpolating subdivision). The scaling property expressed in (2.15) could be expressed as a sentence as follows: "if f belongs to V_j, then a stretched version of f (by a factor 2) belongs to V_{j-1}", which roughly means that $f(2^{-1}t)$ defines an approximation at a coarser resolution. The last condition concerning the existence of a Riesz basis of V_0 implies that there exits $A > 0$ and $B > A$ such that any $f \in V_0$ can be uniquely decomposed as follows

$$f(t) = \sum_{l \in \mathbb{Z}} b_l \theta(t - l),$$

and

$$A\|f\|^2 \leq \sum_{l \in \mathbb{Z}} b_l^2 \leq B\|f\|^2.$$

A simple example of a multiresolution approximation, in the sense of Definition 7 is to consider piecewise constant approximation. The basis θ is the unit box $\theta = \mathbf{1}_{[0,1)}$, and V_j is formed by functions that are constant over $[k2^j, (k+1)2^j)$, $\forall k \in \mathbb{Z}$. Although this example meets the requirements of multiresolution anaysis, one will often prefer to have smooth basis functions.

When constructing *orthonormal* bases of the spaces V_j, it comes out that those bases are built using dilated and translated versions of a single function ϕ, called *scaling function.* The basis of V_j is $\{\phi^j_k\}_{k \in \mathbb{Z}}$ where

$$\phi^j_k(t) = \frac{1}{\sqrt{2^j}} \phi\left(\frac{t - k2^{-j}}{2^{-j}} \right).$$

The scaling function ϕ verifies a scaling equation, inferred from equations (2.14) and (2.15): the function $\frac{1}{\sqrt{2}}\phi(2^{-1}t)$ belongs to V_{-1}, and $V_{-1} \subset V_0$. Decomposing such a function on V_0 yields

$$\frac{1}{\sqrt{2}}\phi(2^{-1}t) = \sum_{l\in\mathbb{Z}} b_l \phi(t-l),$$

which is equivalent to the relation obtained in (2.12). In [19], Daubechies and Lagarias studied the existence and global regularity of *two-scale lattice difference equations*

$$f(x) = \sum_{l=1}^{N} a_l f(nx-l),\ n \in \mathbb{N},\ n \geq 2, \tag{2.18}$$

where n is the arity of the associated scheme. This class of equations include the ones derived in (2.12), when dealing with subdivision schemes having a *finite support*, which is the the case for the majority of subdivision schemes. This study proved that a non-trivial $L^1_{\mathbb{R}}$ solution to the equation may exist only if $\sum_{l=0}^{N} a_l \geq n$. Moreover, if any solution to the equation exists, it will be of compact support $\left[0, \frac{N}{n-1}\right]$ only if

$$\sum_{l=0}^{N} a_l = n. \tag{2.19}$$

It is clear, from Tables 2.1, 2.2 and 2.3 that the examples studied in Section 2.2 all meet the requirement of equation (2.19). In fact, this constraint is a corollary of a more specific theorem, proved by N. Dyn in [35]. Recently, Wang [133] studied the conditions of convergence of schemes associated with refinement equations having positive coefficients.

Theorem 3. *Let us consider a stationary n-ary scheme, with a mask $\{a_k\}$. If the scheme is convergent, then*

$$\sum_{l\in\mathbb{Z}} a_{nl+k} = 1 \quad \forall k \in \{0,\ldots,n-1\}.$$

In the case of a binary scheme, this yields

$$\sum_{k\in\mathbb{Z}} a_{2k} = \sum_{k\in\mathbb{Z}} a_{2k+1} = 1.$$

A few additional properties can be derived in the case of a binary uniform interpolating scheme. We will assume that the grids are regular, i.e. $x_k^j = k2^{-j}$. The space of polynomials being at most of degree p will be denoted by π_p. The property of *polynomial reproduction* and the smoothness of the limit function are connected through the following theorem.

Theorem 4. *An interpolatory scheme that produces $\mathcal{C}^p$ limit functions reproduces polynomials in π_p.*

Proof. Cf. for instance [63], pp. 27-28. □

If we consider $X_0 = \mathbb{Z}$ and $f_k^0 = {x_k^0}^p = k^p$, the above theorem implies that the limit function verifies $\mathcal{S}f_0(x) = x^p,\ \forall x$. It is sufficient to check the polynomial reproduction property for the monomial x^p, since the scheme is linear. As an example, the four-point scheme is exact for cubic polynomials if $w = \frac{1}{16}$. However, as shown in Section 2.3.2 and Section 2.3.4, this scheme only produces $\mathcal{C}^1$ limit functions. Another theorem connects the polynomial reproduction property with the approximation order of the scheme.

Theorem 5. *An convergent interpolating scheme that reproduces polynomials from π_p has an approximation order of $p+1$*

Proof. Cf. for instance [63], pp. 34-35. ☐

Although the above criteria are useful to infer the behavior of the scheme, regarding its convergence and the smoothness of the limit function, they may not be sufficient. While a direct analysis can be performed, the representation of schemes using *Laurent polynomials*, described in the next section, is much easier to handle.

2.3.2 Analysis of linear stationary schemes using Laurent polynomials

The idea of associating a Laurent polynomial to a linear scheme has been introduced and exploited in [10] and [113, 114]. Let us consider a scheme with a mask $\{a_k\}$. The associated polynomial is

$$a(z) = \sum_{k\in\mathbb{Z}} a_k z^k, \ z \in \mathbb{C}.$$

If we denote by $F^j(z) = \sum_{k\in\mathbb{Z}} f_k^j z^k$ the z-transform of f^j, $F^{j+1}(z)$ verifies the following relation

$$F^{j+1}(z) = a(z)F^j(z^n). \tag{2.20}$$

This polynomial has interesting properties, which are stated in the following theorem.

Theorem 6. *Let us consider a n-ary convergent scheme of mask $\{a_k\}$. The associated polynomial $a(z)$ verifies the following properties:*

$$\begin{array}{rcl} a(1) & = & n, \\ a(\zeta_n^p) & = & 0, \ \zeta_n^p = e^{\frac{2ip\pi}{n}}, \ p \in \{1,\dots,n-1\}. \end{array}$$

Proof. The first part is trivially inferred from theorem 3 or from equation (2.19).
Let us show the second part of the theorem. If the scheme is convergent, theorem 3 yields

$$\sum_{l\in\mathbb{Z}} a_{nl+k} = 1 \quad \forall k \in \{0,\dots,n-1\}.$$

Let us rewrite $a(z)$ appropriately

$$a(z) = \sum_{k\in\mathbb{Z}} a_k z^k = \sum_{k=0}^{n-1}\sum_{l\in\mathbb{Z}} a_{nl+k} z^{nl+k}.$$

If we evaluate the $\sum_{l\in\mathbb{Z}} a_{nl+k} z^{nl+k}$ at $z = \zeta_n^p$, we obtain $\zeta_n^{pk}\sum_{l\in\mathbb{Z}} a_{nl+k} = \zeta_n^{pk}$, according to theorem 3. Therefore,

$$a(\zeta_n^p) = \sum_{k=0}^{n-1} \zeta_n^{pk} = 0 \ \forall p \in \{1,\dots,n-1\},$$

since the ζ_n^p are n-th roots of the unity. ☐

Corollary. *There exists a polynomial $b(z)$ such that*

$$a(z) = b(z)\frac{1}{n}\frac{z^n-1}{z-1} = b(z)\frac{1}{n}\prod_{p=1}^{n-1}(z-\zeta_n^p), \ \textit{with } b(1) = n. \tag{2.21}$$

Proof. The existence is justified by the fact that ζ_n^p are roots of $a(z)$. Since $a(1) = n$, we have $b(1) = n$ and

$$\begin{aligned}\prod_{p=1}^{n-1}(1-\zeta_n^p) &= \left(\frac{z^n-1}{z-1}\right)_{z=1}, \\ &= \left(\sum_{k=0}^{n-1} z^k\right)_{z=1} \\ &= n.\end{aligned}$$

□

The application of these properties to a Laurent polynomial associated to a convergent binary scheme, yields

$$a(-1) = 0,\ a(1) = 2,$$

which implies that there exists a polynomial $b(z)$ such that

$$a(z) = \frac{1+z}{2}b(z),\ \text{with } b(1) = 2.$$

For instance, in the case of the four-point scheme, we have

$$a(z) = 1 + \left(\frac{1}{2} + w\right)(z + z^{-1}) - w(z^3 + z^{-3}),$$

which yields

$$b(z) = -2wz^{-3} + 2wz^{-2} + z^{-1} + 1 + 2wz - 2wz^2.$$

The Laurent polynomial $b(z)$ can be seen as another subdivision scheme, related to the initial scheme through the following theorem.

Theorem 7. *Let us denote respectively by $\mathcal{S}_a$ and $\mathcal{S}_b$ the schemes having $a(z)$ and $b(z)$ as associated Laurent polynomials. Let us also define the difference operator Δ as follows*

$$\Delta f^j = \{f_k^j - f_{k-1}^j,\ k \in \mathbb{Z}\}.$$

If $\mathcal{S}_a$ is a convergent scheme,

$$\Delta(\mathcal{S}_a f) = \frac{1}{n}\mathcal{S}_b \Delta f.$$

Proof. Let us denote respectively by $F^j(z)$, and $D^j(z)$ the z-transform of f_j and Δf^j. Clearly, these two polynomials are linked through the following relation

$$D^j(z) = (1-z)F^j(z).$$

Writing the z transform of $D^{j+1}(z)$, using equation (2.21), we have

$$D^{j+1}(z) = (1-z)F^{j+1}(z) = (1-z)a(z)F^j(z^n).$$

If S_a is convergent, $a(z)$ can be written $\frac{1}{n}\frac{1-z^n}{1-z}b(z)$ according to Theorem 6, which yields

$$D^{j+1}(z) = \frac{1-z^n}{n}b(z)F^j(z^n) = \frac{1}{n}b(z)D^j(z^n).$$

□

Note that this theorem is not restricted to regular grids. A similar "commutation formula" has been derived in the irregular case in [17] and [18]. If the scheme is convergent, it is clear that $\lim_{j\to+\infty} \Delta f^j = 0$. Although less intuitive, the converse is also true, which provides a necessary and sufficient condition for a scheme to converge, expressed in the following theorem.

Theorem 8. *The subdivision scheme* $\mathcal{S}_a$ *is convergent if and only if*

$$\lim_{p\to+\infty} (\mathcal{S}_b)^p f_0 = 0,$$

for any initial data f_0.

Proof. (cf. [63, pp. 53-54] which gives the proof for the binary case, from which the current proof is adapted).

In order to prove this theorem, we will show that the piecewise function obtained through subdivision satisfies the conditions of theorem 1, i.e. that there exists $\beta > 0$ and $0 < \alpha < 1$ such that

$$\sup_{t\in\mathbb{R}} |f^{j+1}(t) - f^j(t)| \leq \beta\alpha^j.$$

It is clear that the maximal values of the function $f^{j+1} - f^j$ are reached at the breakpoints, i.e. for binary subdivision

$$\sup_{t\in\mathbb{R}} |f^{j+1}(t) - f^j(t)| = \sup_{k\in\mathbb{Z}} \left\{ \max_{p\in\{0,\ldots,n-1\}} \left| f^{j+1}_{nk+p} - \frac{1}{n}\left((n-p)f^j_k + pf^j_{k+1}\right) \right| \right\}. \tag{2.22}$$

Let us define the function g^{j+1} by

$$g^{j+1}_{nk+p} = \frac{1}{n}\left[(n-p)f^j_k + pf^j_{k+1}\right],\ p \in 0,\ldots,n-1,\ \forall k \in \mathbb{Z},$$

and let us denote respectively by $G^{j+1}(z)$ and $F^j(z)$ the z-transform of g^{j+1} and f^j. F^j and G^{j+1} are linked through the following relation

$$G^{j+1}(z) = m(z)F^j(z^n),\ \text{with}\ m(z) = \frac{z^{-(n-1)}}{n}\left(\frac{1-z^n}{1-z}\right)^2.$$

Using this relation and equation (2.20), we have

$$F^{j+1}(z) - G^{j+1}(z) = F^j(z^2)\left(a(z) - m(z)\right).$$

Writing $a(z)$ as $\frac{1}{n}\frac{1-z^n}{1-z}b(z)$, we have

$$F^{j+1}(z) - G^{j+1}(z) = \frac{1}{n}\frac{1-z^n}{1-z}\left(b(z) - z^{-(n-1)}\frac{1-z^n}{1-z}\right)F^j(z^n).$$

Since $b(z) - z^{-(n-1)}\frac{1-z^n}{1-z} = 0$ for $z = 1$, this quantity can be written as $n(1-z)p(z)$, which yields

$$F^{j+1}(z) - G^{j+1}(z) = p(z)(1-z^n)F^j(z^n).$$

Using the definition of Δ done in theorem 7, if we denote by $D^j(z)$ the z-transform of Δf^j, the following relation holds

$$D^j(z) = (1-z)F^j(z),$$

which yields $D^j(z^n) = (1-z^n)F^j(z^n)$. Thus,

$$F^{j+1}(z) - G^{j+1}(z) = p(z)D^j(z^n). \tag{2.23}$$

Using the infinite norm $\|.\|_\infty$ of the polynomials, defined as $\left\|F^j(z)\right\|_\infty = \max_{k\in\mathbb{Z}} |f_k^j|$, it follows from equation (2.22) that

$$\sup_{t\in\mathbb{R}} |f^{j+1}(t) - f^j(t)| = \left\|F^{j+1}(z) - G^{j+1}(z)\right\|_\infty .$$

Combining this relation with equation (2.23), it comes

$$\begin{aligned}\sup_{t\in\mathbb{R}} |f^{j+1}(t) - f^j(t)| &= \left\|F^{j+1}(z) - G^{j+1}(z)\right\|_\infty ,\\ &= \left\|p(z)D^j(z^n)\right\|_\infty ,\\ &\leq \|p(z)\|_1 \max_{k\in\mathbb{Z}} |f_k^j - f_{k-1}^j|,\\ &\leq \|p(z)\|_1 \left\|\Delta f^j\right\|_\infty ,\\ &\leq \|p(z)\|_1 \left\|n^{-j} S_b^j \Delta f^0\right\| .\end{aligned}$$

Now if we have $\lim_{k\to+\infty}(S_b)^k f^0 = 0$ for any f^0, this implies that $n^{-j}S_b^j$ is contractive, there exists $l \in \mathbb{N}^*$ and $0 < \varepsilon < 1$ such that

$$\left\|\left(\frac{1}{n}S_b\right)^l\right\|_\infty = \varepsilon < 1.$$

For $j \geq l$, this leads to

$$\sup_{t\in\mathbb{R}} |f^{j+1}(t) - f^j(t)| \leq \|p(z)\|_1 \varepsilon^{j/l} \max_{0\leq k<l} \left\|\Delta f^k\right\|_\infty \leq \beta\alpha^j,$$

with $\alpha = \varepsilon^{1/l} < 1$. □

According to theorem 8, checking if a scheme S_a converges is equivalent to check whether we have $\left\|\left(\frac{1}{n}S_b\right)^l\right\|_\infty < 1$ for some l or not. This theorem is also the key to prove higher order regularity of the limit function. Since the condition on the norm of S_b guarantees the uniform convergence of the sequence of functions f_j, it also guarantees the continuity of the limit function. The condition of $\mathcal{C}^l$ continuity is expressed in the following theorem

Theorem 9. *Let us consider a scheme, with associated Laurent polynomial $a(z)$. If there exists a polynomial $b(z)$ such that*

$$a(z) = \left(\frac{1}{n}\frac{z^n-1}{z-1}\right)^l b(z),$$

where $l \in \mathbb{N}^*$, and such that the associated scheme $n^{-1}S_b$ is contractive, the limit function is $\mathcal{C}^l$ for any initial set.

For binary subdivision, the relation between a and b can be written as $a(z) = \frac{(1+z)^l}{2^l} b(z)$. A simple example of scheme meeting those requirements is to consider the box-spline of order p from equation (2.7). Clearly, the box-spline of order p leads to $\mathcal{C}^{p-1}$ limit functions.

The infinite norm of a scheme, used in the above theorems is defined using the refinement rule in equation (2.3)

$$\|Sf\|_\infty \leq \max_{k\in\mathbb{Z}} \sum_{l\in\mathbb{Z}} |\alpha_{k-nl}| \, \|f\|_\infty ,$$

which yields (cf. sec.3.3 in [10])

$$\|S\|_\infty = \max_{k\in\mathbb{Z}} \left\{\sum_{l\in\mathbb{Z}} |\alpha_{k-nl}|\right\} . \tag{2.24}$$

In the case of a binary scheme, we have

$$\|S\|_\infty = \max\left\{\sum_{l\in\mathbb{Z}} |\alpha_{2i}|,\ \sum_{l\in\mathbb{Z}} |\alpha_{2i+1}|\right\}.$$

The computation of $\|n^{-l}S_b^l\|_\infty$ is done using equation (2.20). According to this relation, the mask of S_b^l is given by the polynomial

$$b_l(z) = b(z)b(z^n)b(z^{n^2})\dots b(z^{n^{l-1}}).$$

Let us denote by $\{b_{l,k}\}_{k\in\mathbb{Z}}$ the coefficients of $b_l(z)$. Using theorem 7, we have

$$D^{j+l}(z) = b(z)D^{j+l-1}(z^n) = \cdots = b_l(z)D^j\left(z^{n^l}\right),$$

which is equivalent to the following refinement rule

$$(\Delta f^{j+l})_k = \sum_p b_{l,k-pn^l}(\Delta f^j)_p.$$

According to equation (2.24), it follows

$$\|S_b^l\|_\infty = \max_{0\le k<n^l}\left\{\sum_p |b_{l,k-pn^l}|\right\}. \tag{2.25}$$

2.3.3 Illustration of the Laurent polynomial technique

Let us consider two of the subdivision schemes presented in Section 2.2, namely the corner cutting scheme and the binary four-point scheme.

1. *Corner-cutting*

   Let us consider a corner cutting rule, as defined in equation (2.6). The associated polynomial is

   $$a(z) = \beta_1 z^{-2} + (1-\beta_2)z^{-1} + (1-\beta_1) + \beta_2 z.$$

   We have to check the contractivity of the scheme $2^{-1}S_b$ given by

   $$\frac{1}{2}b(z) = \beta_1 z^{-2} + (1-\beta_1-\beta_2)z^{-1} + \beta_2.$$

   The convergence conditions are

   $$\begin{aligned} |\beta_1| + |\beta_2| &< 1, \\ |1-\beta_1-\beta_2| &< 1, \end{aligned}$$

   which are automatically met under the initial assumptions ($\beta_1 > 0$, $\beta_2 > 0$ and $\beta_1 + \beta_2 < 1$). This proves the convergence to a continuous function (or curve) for the de Rham scheme ($\beta = \frac{2}{3}$) and for Chaikin's rule ($\beta = \frac{3}{4}$). In order to check $\mathcal{C}^1$ continuity of the limit function, we have to prove that S_b converges. The first requirement is $b(-1) = 0$, which implies $\beta_1+\beta_2 = \frac{1}{2}$. We also have to check that the scheme having for polynomial

   $$2^{-1}c(z) = \frac{b(z)}{1+z} = 2\beta_1 z^{-2} + 2\left(\frac{1}{2} - \beta_1\right)z^{-1},$$

   is contractive, which is achieved for $0 < \beta_1 < \frac{1}{2}$. This proves that Chaikin's corner cutting is $\mathcal{C}^1$ ($\beta_1 = \beta_2 = \frac{1}{4}$), whereas de Rham's trisection is not ($\beta_1 = \beta_2 = \frac{1}{3}$).

2. *Binary four-point scheme*

   We have
$$b(z) = -2wz^{-3} + 2wz^{-2} + z^{-1} + 1 + 2wz - 2wz^2,$$
   which leads to
$$\|2^{-1}S_b\|_\infty = \frac{1}{2} + 2|w|.$$
   Thus, the convergence condition stated in theorem 8 implies $|w| < \frac{1}{4}$. The bound on $\left\|2^{-2}S_b^2\right\|_\infty$ is less restrictive than this one. In fact, if performing the analysis for all powers of S_b, the scheme converges for $|w| < \frac{1}{2}$.The $\mathcal{C}^1$-continuity depends on the convergence of the scheme having
$$\frac{1}{2}c(z) = \frac{b(z)}{1+z} = -2wz + 4w + (1-4w)z^{-1} + 4wz^{-2} - 4w^{z}{-}3,$$
   as associated Laurent polynomial. However, in this case, we have
$$\left\|2^{-1}S_c\right\|_\infty = \max\{8|w|, |1-4w| + 4|w|\} \geq 1.$$
   We cannot conclude from this whether the scheme converges or not, which implies that we have to consider $\left\|2^{-2}S_c^2\right\|_\infty$. After computation, we have
$$\left\|2^{-2}S_c^2\right\|_\infty = \max\Big\{16w^2 + 4|w|,\ 4|w| + |1-8w|,\ 8w^2 + 4|w||1-2w|\Big\}.$$
   Thus, the condition $\left\|2^{-2}S_c^2\right\|_\infty < 1$ yields the following constraints
$$\begin{aligned} \frac{1-\sqrt{5}}{8} < \quad & w \quad < \frac{\sqrt{5}-1}{8}, \\ 0 < \quad & w \quad < \frac{1}{6}, \\ \frac{1-\sqrt{5}}{8} < \quad & w \quad < \frac{1}{4}. \end{aligned}$$
   Taking the most restrictive conditions yield $0 < w < \frac{\sqrt{5}-1}{8}$ to achieve $\mathcal{C}^1$-convergence of the four-point scheme. This bound was derived by Dyn and Levin [37], by studying finite-differences and using the eigenanalysis detailed in Section 2.3.4. The initial estimate given in [36] was $0 < w < \frac{1}{8}$ to achieve $\mathcal{C}^1$ convergence. According to the polynomial reproduction property from theorem 4, the four-point scheme can only be $\mathcal{C}^2$ for $w = \frac{1}{16}$. In order to check $\mathcal{C}^2$-convergence of the scheme, we have to check whether $\left\|2^{-l}S_d^l\right\|_\infty < 1$ for some l, with
$$\frac{1}{2}d(z) = \frac{c(z)}{1+z} = \frac{1}{4}(-z^{-3} + 3z^{-2} + 3z^{-1} - 1).$$
   Clearly, we have $\left\|2^{-1}S_d\right\|_\infty = 1$. In fact, for any l, $\left\|2^{-l}S_d^l\right\|_\infty = 1$, which does not permit to make any conclusion regarding the $\mathcal{C}^2$ continuity of the limit function. To lift the indetermination, it is necessary to introduce the concept of *eigenanalysis*, developed in the next section.

2.3.4 Eigenanalysis of stationary schemes

In [20] and [37], the refinement relation from equation (2.18) was analyzed using two associated refinement matrices T_0 and T_1, defined by
$$\begin{aligned} (T_0)_{i,j} &= a_{2i-j-1},\ 1 \leq i,j \leq N, \\ (T_1)_{i,j} &= a_{2i-j},\ 1 \leq i,j \leq N. \end{aligned}$$

The above definitions are valid for binary schemes. In the case of n-ary subdivision, there are $n-1$ of such matrices. In the case of the four-point scheme (for $w = \frac{1}{16}$), the matrices are

$$
\boldsymbol{T}_0 = \begin{pmatrix}
-\frac{1}{16} & 0 & 0 & 0 & 0 & 0 \\
\frac{9}{16} & 0 & -\frac{1}{16} & 0 & 0 & 0 \\
\frac{9}{16} & 1 & \frac{9}{16} & 0 & -\frac{1}{16} & 0 \\
-\frac{1}{16} & 0 & \frac{9}{16} & 1 & \frac{9}{16} & 0 \\
0 & 0 & -\frac{1}{16} & 0 & \frac{9}{16} & 1 \\
0 & 0 & 0 & 0 & -\frac{1}{16} & 0
\end{pmatrix},
$$

$$
\boldsymbol{T}_1 = \begin{pmatrix}
0 & -\frac{1}{16} & 0 & 0 & 0 & 0 \\
1 & \frac{9}{16} & 0 & -\frac{1}{16} & 0 & 0 \\
0 & \frac{9}{16} & 1 & \frac{9}{16} & 0 & -\frac{1}{16} \\
0 & -\frac{1}{16} & 0 & \frac{9}{16} & 1 & \frac{9}{16} \\
0 & 0 & 0 & -\frac{1}{16} & 0 & \frac{9}{16} \\
0 & 0 & 0 & 0 & 0 & -\frac{1}{16}
\end{pmatrix}.
$$

If the conditions of theorem 3 are satisfied, this implies that the vector $\boldsymbol{e}_1 = (1\ 1 \ldots 1)$ is a left eigenvector of $\boldsymbol{T}_0$ and $\boldsymbol{T}_1$, with eigenvalue 1.

Theorem 10. *A sufficient condition for the scheme to converge toward a continuous limit function is that there exists $\lambda < 1$ and $C > 0$ such that for all $m \in \mathbb{N}$,*

$$
\max_{\substack{d_k = 0 \ \textit{or} \ 1 \\ k=1,\ldots,m}} \| \boldsymbol{T}_{d_0} \boldsymbol{T}_{d_1} \ldots \boldsymbol{T}_{d_m} |_{E_1} \| \leq C \lambda^m, \tag{2.26}
$$

where E_1 is the $N-1$-dimensional subspace orthogonal to $\boldsymbol{e}_1$. The limit function f is Hölder continuous,

$$
|f(x) - f(x)| \leq C |x - y|^{\alpha},
$$

with $\alpha = \frac{|\ln \lambda|}{\ln 2}$.

Proof. Cf. the demonstration of theorem 2.2 in [20]. □

The condition expressed in equation (2.26) is equivalent to a condition on the joint spectral radius of $\boldsymbol{T}_0|_{E_1}$ and $\boldsymbol{T}_1|_{E_1}$, denoted by $\rho(\boldsymbol{T}_0|_{E_1}, \boldsymbol{T}_1|_{E_1})$, namely

$$
\rho(\boldsymbol{T}_0|_{E_1}, \boldsymbol{T}_1|_{E_1}) = \lim_{m \to \infty} \sup \left[\max_{\substack{d_k = 0 \text{ or } 1 \\ k=1,\ldots,m}} \| \boldsymbol{T}_{d_0} \boldsymbol{T}_{d_1} \ldots \boldsymbol{T}_{d_m} |_{E_1} \|^{\frac{1}{m}} \right] < 1.
$$

This condition is less easy to handle than the conditions on Laurent polynomials presented in the previous section. However, a necessary condition for equation (2.26) to hold is that 1 is a simple eigenvalue for both $\boldsymbol{T}_0$ and $\boldsymbol{T}_1$.

Using theorem 10, additional to achieve $\mathcal{C}^l$ continuity of the limit function. First, the scheme must satisfy the following *sum rules*

$$
\sum_{k=0}^{N} (-1)^n k^p c_k = 0, \ \forall p \in \{0,\ 1, \ldots L\}. \tag{2.27}
$$

These rules are equivalent to the necessary condition on the associated Laurent polynomial $a(z)$, which has to be divisible by $(1+z)^{L+1}$ for the scheme to converge toward a $\mathcal{C}^L$ limit function.

Theorem 11. *A sufficient condition for a binary scheme to converge toward a $\mathcal{C}^l$ function is that the mask satisfies L sum rules as in equation (2.27), and that there exist $\frac{1}{2} \leq \lambda < 1$, $0 \leq l \leq L$ and $C > 0$, such that for all $m \in \mathbb{N}$*

$$\max_{\substack{d_k=0 \text{ or } 1\\ k=1,\ldots,m}} \|\boldsymbol{T}_{d_0}\boldsymbol{T}_{d_1}\ldots\boldsymbol{T}_{d_m}|_{E_{L+1}}\| \leq C\lambda^m 2^{-lm}, \tag{2.28}$$

where the space E_k is defined as the subspace of $\mathbb{R}^N$ orthogonal to $U_k = span\{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_k\}$, with $\boldsymbol{u}_k = (1^{k-1}, 2^{k-1}, \ldots, N^{k-1})$. Moreover, if $\lambda > \frac{1}{2}$, the l-th derivative of the limit function is Hölder continuous with exponent $\alpha \geq \frac{|\ln\lambda|}{\ln 2}$. If $\lambda = \frac{1}{2}$, the l-th derivative satisfies

$$|f^{(l)}(x+t) - f^{(l)}(x)| \leq C|t||\ln|t||. \tag{2.29}$$

Proof. Cf. [20], the proof of theorem 3.1. □

An interesting intermediate lemma used to prove this theorem, also shown in [37], states that the sum rules from equation (2.27) and equation (2.28) can only hold if the $l+1$ eigenvalues $1, \frac{1}{2}, \ldots, 2^{-l}$ are all simple and all other eigenvalues γ satisfy $|\gamma| < 2^{-l}$.

In the case of the four-point scheme, the matrices $\boldsymbol{T}_0$ and $\boldsymbol{T}_1$ have 1 and $\frac{1}{2}$ as simple eigenvalues. $\frac{1}{4}$ is a double, degenerate eigenvalue, thus the scheme can only produce $\mathcal{C}^1$ limit functions. Such a result was suspected from the analysis using Laurent polynomials. In fact, it can be shown that the derivative satisfies equation (2.29), which implies that it is Hölder continuous with all exponents $\varepsilon > 0$, hence the notation of $\mathcal{C}^{2-\varepsilon}$ to express the regularity of the limit functions produced by the four-point scheme.

2.4 Analysis of non-stationary/non-uniform schemes

Although their interest may not appear at first sight, non-stationary or non-uniform schemes can be useful, since they provide more flexibility and lead to special properties, such as (complex) exponential values reproduction, which are desirable for certain classes of applications. The techniques presented in Section 2.3 form the bases to study non-stationary [14, 42, 43, 44] and non-uniform subdivision [38, 86]. This section briefly presents the main results concerning non-stationary and non-uniform subdivision.

2.4.1 Non-stationary schemes

The theoretical bases regarding non-stationary schemes are derived from the analysis of stationary schemes presented in Sections 2.3.2 and 2.3.4. The analysis of non-stationary schemes has been done in [42, 43] but in a much more detailed way in [44]. Another approach was to study generalized refinement equations with continuous masks is done in [28], and a multiresolution-oriented study of such schemes can be found in [14]. Under certain conditions, the analysis of non-stationary schemes is close to the stationary case. In this section, we summarize the main results regarding the analysis of non-stationary subdivision schemes.

Let us denote by $\{a_k^j\}_{k\in\mathbb{Z}}$ the mask of the scheme at level j. A sufficient condition to achieve $\mathcal{C}^0$ convergence is to have each scheme to verify the rule from equation (2.21), and the associated difference scheme b_j to be contractive. One other tool to analyze is to consider an *asymptotically equivalent scheme.*

Definition 8. *Two (non-stationary) schemes,* $\{S_{a_j}\}$ *and* $\{S_{b_j}\}$ *are* ***asymptotically equivalent*** *if*

$$\sum_{j\in\mathbb{N}} \left\|S_{a_j} - S_{b_j}\right\|_\infty < \infty,$$

where the infinite norm is defined by equation (2.24)*.*

An interesting criteria for non-stationary schemes is stability, defined as follows.

Definition 9. *A subdivision scheme* $\{S_{a_j}\}$ *is* ***stable*** *if there exists* $M \in \mathbb{R}$ *such that*

$$\sup_{\substack{j\in\mathbb{N}\\ m\in\mathbb{N}}} \left\|S_{a_{j+m}} \dots S_{a_j}\right\|_\infty \leq M.$$

The stability is transmitted through the asymptotic equivalence, i.e. if two schemes are asymptotically equivalent, one is stable if and only if the other is also stable. Using these definitions, the following theorem links the convergence with those criteria.

Theorem 12. *(Convergence of asymptotically equivalent schemes)*

1. *If* $\{S_{a_j}\}$ *and* $\{S_{b_j}\}$ *are asymptotically equivalent, the scheme* $\{S_{a_j}\}$ *is* $\mathcal{C}^0$ *and stable if and only if* $\{S_{b_j}\}$ *is* $\mathcal{C}^0$ *and stable.*

2. *If* $\{S_{b_j}\}$ *is stationary and of finite-support, and converges toward a continuous limit function, then* $\{S_{a_j}\}$ *is* $\mathcal{C}^0$ *and stable.*

Proof. Cf. [44], theorem 7. □

Using this theorem, we can use a stationary equivalent scheme in order to prove the convergence of a non-stationary scheme, under the assumption of asymptotic equivalence, which is not a too drastic requirement. This can be extended to check $\mathcal{C}^l$ continuity of a non-stationary scheme through the following theorem.

Theorem 13. *Let us consider two asymptotically equivalent schemes of finite-support* $\{S_{a_j}\}$ *(non-stationary) and* S_a*. If* S_a *leads to* $\mathcal{C}^l$ *limit functions and the following term*

$$\sum_{j\in\mathbb{N}} 2^{lj} \left\|S_{a_j} - S_a\right\|_\infty,$$

remains finite, then $\{S_{a_j}\}$ *is also* $\mathcal{C}^l$*.*

Of course, not all schemes have such a nice equivalent. One counter-example is the scheme given by $a_j(z) = 2^{-(j+1)}(1+z)^j$, which leads to $\mathcal{C}^\infty$ limit functions. The limit function and the associated functional equation has been studied in [116]. An example of non-stationary scheme satisfying the conditions for the above convergence theorem is the "exponential" scheme defined by

$$a_j(z) = \frac{1}{2}(1 + e^{x2^{-(j+1)}} z)b(z),\ x \in \mathbb{R},$$

with $b(z)$ defining (at least) a convergent scheme. This scheme is asymptotically equivalent to $a(z) = \frac{1+z}{2}b(z)$.

2.4.2 Non-uniform schemes

A natural extension to non-stationary subdivision is to consider *non-uniform* subdivision, i.e. to have for each sample f_k^j an associated mask $a_k^j(z)$. The generalization of the Laurent polynomials technique to such schemes has been done in [86], proposing easy to check regularity criteria for such schemes, although the necessary and sufficient conditions might be more restrictive. Before this study, the case of non-uniform corner-cutting had been addressed in [55] (although published in 1996, the paper was written in the end of the 1980 years). In this paragraph, we will focus on the study using the associated Laurent polynomials to each scheme. We will also limit ourselves to binary schemes, although similar properties can be derived for n-ary schemes.

A first assumption, which is required for the uniform case, is that all polynomials satisfy equation (2.19) and also satisfy the conditions in Theorem 6. Another reasonable assumption is that all $\{a_k^j\}$ have a finite-support.

The approach followed in [86] is to compute the polynomials generating values at level $j+l$ from the level j, denoted by $a_k^{j,l}(z)$ defined by the following recursion

$$a_k^{j,p+1}(z) = \sum_m (a_k^{j+p,1})_m a_p^{j,\frac{k-m}{2}}\left(z^2\right) z^m,$$

where $(a_k^j)_m$ denotes the coefficient of the term of degree m from the polynomial $a_k^j(z)$ and $a_{k,1}^j(z) = a_k^{j+1}(z)$. The definition of contractivity for a non-uniform scheme $\{a_k^j(z)\}$ follows from the above relation. A scheme is contractive is there exists $l \in \mathbb{N}$ and $0 \leq \varepsilon < 1$ such that

$$\sum_p |(a_{k,l}^j)_{2^l p+r}| \leq \varepsilon, \text{ with } 0 \leq r < 2^l, \tag{2.30}$$

holds for all k and for j big enough.

Hence, the sufficient condition for $\mathcal{C}^l$ continuity of the limit function, similar to Theorem 8 defined using the finite differences of associated polynomials, stated in the following theorem.

Theorem 14. *Let us define the finite difference polynomial of a non-uniform scheme* $\{a_k^j(z)\}$ *as follows*

$$\begin{aligned} a_k^{j\,[0]}(z) &= a_k^j(z), \\ a_k^{j\,[p+1]}(z) &= \frac{2^p}{z^2-1}\Big(z a_{k-1}^{j\,[p]}(z) - a_k^{j\,[p]}(z)\Big), \; p > 0. \end{aligned} \tag{2.31}$$

If $\{a_k^{j\,[l+1]}(z)\}$ *is contractive, then the scheme* $\{a_k^j(z)\}$ *is* $\mathcal{C}^l$.

An example of non-uniform is the non-uniform four-point scheme where the parameter w is replaced by w_k^j. The second order finite difference is

$$a_k^{j\,[2]}(z) = z^{-1} - 2(z^2 w_{k-2}^j - 2w_{k-1}^j + w_k^j)z^{-3}(z^2+1).$$

The convergence is checked using equation (2.30). As for the uniform four-point scheme, we have to check the contractivity for $l = 2$. David Levin showed that the $\mathcal{C}^1$ continuity held for any choice of $w_k^j \in \left[\varepsilon, \frac{1}{8} - \varepsilon\right]$ with ε small enough.

Let us consider the example developed in [55], i.e. the non-uniform corner cutting. The subdivision masks are defined by

$$a_k^j(z) = \alpha_k^j z^{-2} + (1-\beta_k^j)z^{-1} + (1-\alpha_k^j) + \beta_k^j z,$$

with $\alpha_k^j > 0$, $\beta_k^j > 0$ and $\alpha_k^j + \beta_k^j < 1$ for all j and k. It is clear that the conditions $a_k^j(1) = 2$ and $a_k^j(-1) = 0$ hold for all j and k. The first order difference is

$$a_k^{j\,[1]}(z) = \alpha_k^j z^{-2} + (1 - \alpha_{k-1}^j - \beta_k^j)z^{-1} + \beta_{k-1}^j,$$

which yields the following conditions for $\mathcal{C}^0$ continuity

$$\begin{aligned} |\alpha_k^j + \beta_{k-1}^j| &< 1, \\ |1 - \alpha_{k-1}^j - \beta_k^j| &< 1, \end{aligned}$$

for all j and k. If moreover we assume that the sets $\{\alpha_k^j\}_{(j,k)\in\mathbb{N}\times\mathbb{Z}}$ and $\{\beta_k^j\}_{(j,k)\in\mathbb{N}\times\mathbb{Z}}$ have a finite upper and lower bounds, denoted by $\bar{\alpha}$, $\underline{\alpha}$, $\bar{\beta}$ and $\underline{\beta}$, the following conditions are sufficient to ensure $\mathcal{C}^0$ continuity

$$\underline{\alpha} > 0,\ \underline{\beta} > 0,\ \bar{\alpha} + \bar{\beta} < 1.$$

The conditions for $\mathcal{C}^1$ continuity are trickier to derive. In order for $za_{k-1}^{j\,[1]}(z) - a_k^{j\,[1]}(z)$ to be divided by $z^2 - 1$, the following conditions are necessary

$$\begin{aligned} \alpha_k^j &= 1 - (2\beta_{k-1}^j + \alpha_{k-2}^j), \\ \beta_k^j &= 1 - (2\alpha_{k-1}^j + \beta_{k-2}^j), \end{aligned}$$

for all j and k. The conditions $\alpha_k^j > 0$, $\beta_k^j > 0$ and $\alpha_k^j + \beta_k^j < 1$ then imply

$$\begin{aligned} 2\beta_{k-1}^j + \alpha_{k-2}^j &< 1, \\ 2\alpha_{k-1}^j + \beta_{k-2}^j &< 1. \end{aligned}$$

Under the assumption of finite upper and lower bounds of $\{\alpha_k^j\}_{(j,k)\in\mathbb{N}\times\mathbb{Z}}$ and $\{\beta_k^j\}_{(j,k)\in\mathbb{N}\times\mathbb{Z}}$, this yields the following necessary constraints on the bounds

$$\begin{aligned} &2\bar{\alpha} + \bar{\beta} < 1, \\ &2\bar{\beta} + \bar{\alpha} < 1. \end{aligned} \tag{2.32}$$

The contractivity of the second order difference is achieved if $\bar{\alpha} + 2\bar{\beta} < 2$ and $\underline{\alpha} + \underline{\beta} > 0$. The first condition is met because of the constraints expressed in equation (2.32), and chosing $\underline{\alpha} > 0$ and $\underline{\beta} > 0$ is an easy way to satisfy the second condition. In fact, the original study in [55] shows that the following conditions are sufficient for the non-uniform corner cutting to converge toward a $\mathcal{C}^1$ limit function

$$\underline{\alpha} > 0,\ \underline{\beta} > 0, \text{ and } 2\bar{\alpha} + \bar{\beta} < 1,\ 2\bar{\beta} + \bar{\alpha} < 1.$$

Not surprisingly, Chaikin's corner cutting rule meets those requirements ($\alpha = \beta = \frac{1}{4}$) whereas de Rham trisection only satisfies the $\mathcal{C}^0$ conditions ($\alpha = \beta = \frac{1}{3}$). In this case, the analysis of the scheme performed with Laurent polynomials leads to much more restrictive conditions for $\mathcal{C}^1$ convergence than the direct analysis conducted in the original study. This approach still provides useful convergence criteria for non uniform schemes.

In any case, dealing with non-uniform schemes is not a trivial operation and requires often restrictive assumptions to be made in order to ensure convergence, as for the non-uniform version of the four-point scheme or non-uniform corner cutting. A non-trivial of such non-uniform schemes is to consider fully irregular (but still linear) subdivision.

For the sake of completeness, we mention also works that have studied subdivision for irregular grids. The problems raised by such schemes have been initially considered by J. Warren in [134] and studied the convergence of schemes performing midpoint insertion on an initial irregular grid. Several theoretical developements regarding fully irregular refinements (i.e. having an initial irregular grid and an irregular refinement) were done by Daubechies, Sweldens and Guskov in [17] (although this study was focused on the irregular Lagrange scheme presented in Section 2.2) and further developed in a sequel [18]. The analysis performed in those papers relies on the analysis of the finite differences of a grid. The proofs rely on a commutation formula, which is the analogous of equation (2.21). Using the schemes associated to the finite difference, it can be shown that convergence and regularity are achieved, under a non-negligible number of assumptions on the local regularity of the grid and the scheme itself. Recently, the polynomial reproduction property of schemes dealing with non-uniform grids has been studied in [85]

Linear subdivision of discrete surfaces

3

Computer graphics and computer-aided design are two consumers of the refinement processes described in Chapter 2. Historically, Bézier curves were designed to create easily a smooth curve from a polygon, and were developed and used in the car industry. A natural extension of the aspects of subdivision that concerned curves was to consider surfaces. In this chapter, we will review surface subdivision techniques. As for the univariate case, we will deal with discrete data. If the relation between a univariate function and its discrete version is trivial, there are several way to represent discrete surfaces, which will be detailed in Section 3.1. A simple generalization of the univariate case to a specific class of discrete surfaces will be detailed in Section 3.2 and the generic subdivision techniques for discrete polygonal surfaces and their analysis will be addressed in Section 3.3.

3.1 Surfaces

In order to fully understand the properties of discrete surfaces, this section will remind the classical definitions of continuous ones. We will present the most common representation of discrete surfaces.

3.1.1 Continuous surfaces

Although not precised in the title, the *surfaces* we implicitly refer to are assumed to be, mathematically speaking, "2-dimensional surfaces in $\mathbb{R}^3$". However, the concept of surfaces is not that restrictive. In fact, the definition of surfaces is enclosed in the definition of a *topological manifold*. We refer the reader to [77] and [98] for more advanced concepts regarding the theory of continuous surfaces and differential geometry.

Definition 10. *A topological manifold $\mathcal{M}$ of dimension n is a separable topological space, having a countable basis $(U_i)_{i\in I}$ of open sets, which are homeomorphic to open sets (V_i) from $\mathbb{R}^n$. The set of functions ϕ_i mapping U_i to V_i is called the atlas, ϕ_i is a local chart, and ϕ_i^{-1} is a local parameterization. A topological manifold is termed* differentiable *(resp. $\mathcal{C}^k$) if for all i and j, the* *application*

$$\phi_{i,j} : \phi_i(U_i \cap U_j) \xrightarrow{\phi_i^{-1}\phi_j} \phi_j(U_i \cap U_j),$$

is differentiable (resp. $\mathcal{C}^k$*).*

The relation with "real-world" surfaces may not appear at first sight from this definition. However, the surfaces we intend to deal with are covered by this definition. For instance, the unit sphere $\mathcal{S}^n = \{x \in \mathbb{R}^{n+1}, \|x\| = 1\}$ is a $\mathcal{C}^\infty$ manifold of dimension n.

From this definition, it is fairly easy to derive a parametric definition of a surface.

Definition 11. *A surface of dimension* m *in* $\mathbb{R}^n$ $(m < n)$ *is defined by an open set* $U \subset \mathbb{R}^m$*, and a mapping*

$$\begin{array}{rccl} \boldsymbol{x} : & U & \longrightarrow & \mathbb{R}^n \\ & (u_1, \ldots, u_m) & \longmapsto & \Big(x_1(u_1, \ldots, u_m), \ldots, x_n(u_1, \ldots, u_m)\Big), \end{array}$$

such that its Jacobian matrix is of rank m*. The regularity of the surface is given by the regularity of* $\boldsymbol{x}$*.*

This definition is still very generic, but we will restrict it to surfaces of dimension 2 in $\mathbb{R}^3$, which leads to

$$\begin{aligned} \boldsymbol{x} : U \subset \mathbb{R}^2 &\longrightarrow \mathbb{R}^3 \\ (u,v) &\longmapsto \begin{pmatrix} x(u,v) \\ y(u,v) \\ z(u,v) \end{pmatrix}. \end{aligned} \tag{3.1}$$

Let us illustrate this by considering a simple example, namely a sphere. Such a surface can be represented simply using the spherical coordinates

$$\begin{aligned} x(\phi,\theta) &= r\cos\theta\sin\phi, \\ y(\phi,\theta) &= r\sin\theta\sin\phi, \\ z(\phi,\theta) &= r\cos\phi, \end{aligned}$$

with $\phi \in [0,\pi]$, $\theta \in [-\pi,\pi]$ and where r is the radius of the sphere. This connection is illustrated in Figure 3.1.

The parameterization of a surface is not unique, and may not be easily accessible in some cases. Another common representation of a surface is achieved through a zero level-set of a function f, i.e. the point M of coordinates (x,y,z) belongs to the surface if $f(x,y,z) = 0$. For instance, the sphere of radius r can be represented by the equation $x^2 + y^2 + z^2 - r^2 = 0$. The parametric and implicit representations are connected through the implicit function theorem.

For this theorem to be applicable, f must be at least $\mathcal{C}^1$. Let us consider a point (x_0, y_0, z_0) belonging to the surface, i.e. $f(x_0,y_0,z_0) = 0$, such that one of the partial derivatives at this point is not zero. We will assume $\frac{\partial f}{\partial z}(x_0,y_0,z_0) \neq 0$. Under such assumptions, there exists a neighborhood U of z_0, a neighborhood V of (x_0,y_0) and a function g, $g : V \longrightarrow U$ at least $\mathcal{C}^1$, such that

$$\forall (x,y,z) \in V \times U,\ f(x,y,z) = 0 \text{ if and only if } z = g(x,y).$$

However, finding g from f may not be trivial, and in some cases, a closed form is not available. There are however techniques that are based on implicit representations of surfaces, referred to as *level-set* techniques. We refer the reader to [121] for a study of this topic.

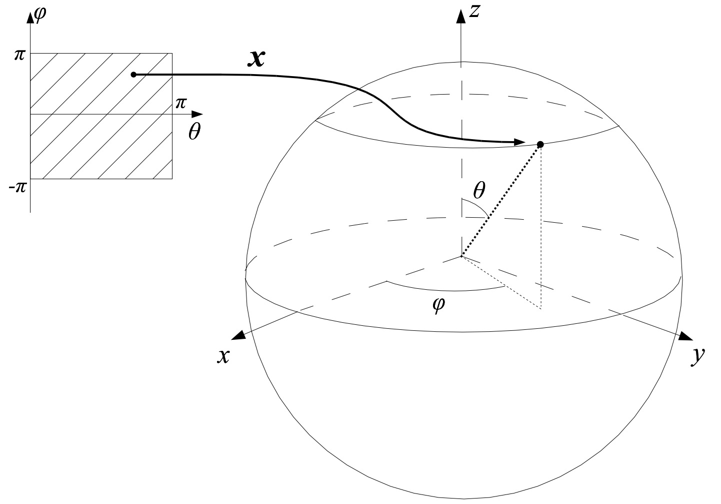


Figure 3.1: Surface - parametric representation

3.1.2 Discrete surface representations

Sampling a univariate function $f(x)$ is a trivial operation, achieved by taking the samples $f_k = f(x_k)$. Adapting this to surfaces having a parametric representation is quite obvious. Using equation (3.1), a natural discretization is to evaluate $\boldsymbol{x}$ for a set of (u_j, v_k).

The problem in this approach is that the access to such a parameterization is not guaranteed. Moreover, real-world surfaces cannot be (in general) turned into equations (implicit or parametric) that easily. Hence, the need for another approach, which is to directly consider sample points, called *vertices*, that belong to the surface (e.g. the data acquired from a laser scanner). By connecting the points together to form planar polygons called *faces* (e.g. triangles), we define a piecewise linear approximation of a surface, using a *polygonal mesh*, as shown for instance in Figure 3.2. The representation of surfaces using polygonal meshes is one of the most generic. It suffers two major drawbacks, namely that a smooth surface (or a surface having relatively small details) will require a large number of vertices and faces to be rendered properly, and that in this case it becomes quite difficult to modify. Nevertheless, this representation is a reference in terms of computer graphics and adopted widely, in hardware and related APIs such as OpenGL [140], but also in standards, e.g. VRML (Virtual Reality Modelling Language) [66, 137] or its successor X3D [138]. We will detail the formalism associated to polygonal meshes in Section 3.3, after the study of a particular case in Section 3.2.

Instead of using many polygons to represent a smooth surface, another approach consists in using a smaller number of smooth patches, having their shape and position controlled by a coarse polygonal mesh (or *control mesh*). The vertices of the control mesh are called *control points*. The control mesh is usually formed by simple polygons, usually quadrilaterals (or triangular). We will assume that the control points $\boldsymbol{p}_{i,j} \in \mathbb{R}^3$ can be grouped in quadrilaterals $(\boldsymbol{p}_{i,j}, \boldsymbol{p}_{i+1,j}, \boldsymbol{p}_{i+1,j+1}, \boldsymbol{p}_{i,j+1})$. The most common approach is to use tensor products to define a surface from this control mesh, i.e.

$$\begin{pmatrix} x(u,v) \\ y(u,v) \\ z(u,v) \end{pmatrix} = \sum_{i=0}^{n} \sum_{j=0}^{m} \boldsymbol{p}_{i,j} P_{i,n}(u) P_{j,m}(v), \tag{3.2}$$

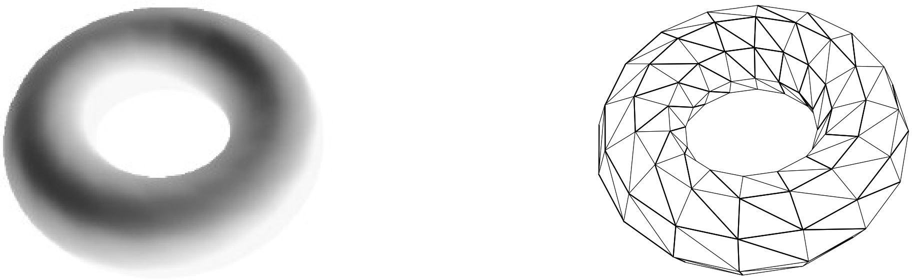


Figure 3.2: Continuous torus (left) and discrete approximation using a polygonal mesh (in this case, a triangular mesh).

where $\{P_{k,r}(x)\}$ is a family of polynomials of degree r. The choice of polynomials is driven by the desired quality of the approximation of the control mesh and complexity. In fact, only a few families of polynomials are practically used, namely Bernstein polynomials (leading to Bézier curves) and spline polynomials. We refer the reader to [50, 93], [63] and [132] for a detailed presentation of Bézier polynomials and B-splines as well as various applications of such techniques. To sum up quickly, these polynomials have common properties, the most important being *local support*, which is useful to edit locally the surface by modifying only a few control points. Another interesting property is the *convex hull* property, i.e. the curve (or surface) always lies in the convey hull formed by the control points. The polynomials of degree n are also usually computed recursively from the polynomials of degree $n-1$, e.g. for the Bernstein polynomials

$$B_{i,n}(t) = \begin{pmatrix} i \\ n \end{pmatrix} t^i (1-t)^{n-i},\ t \in [0,1],$$

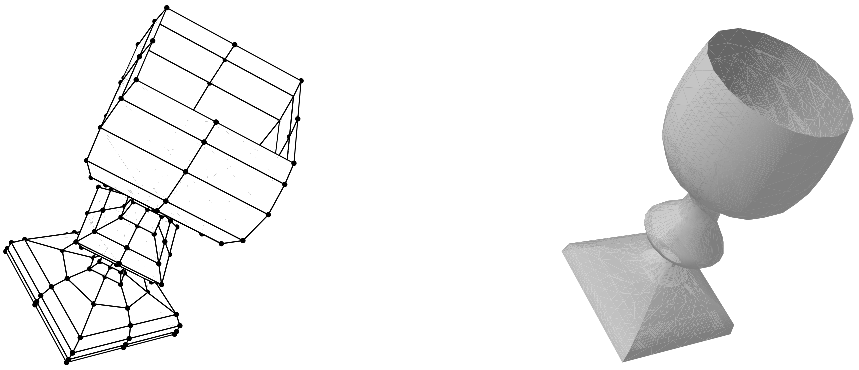
which leads to the following recursion relation (which is a consequence of Pascal's triangle relation)

$$B_{i,n}(t) = (1-t)B_{i,n-1}(t) + tB_{i-1,n-1}(t).$$

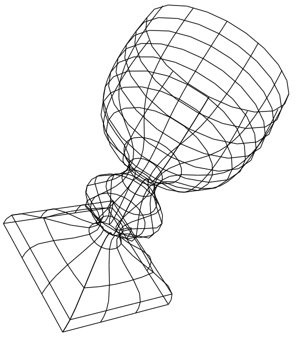
The representation of equation (3.2) is often generalized to *rational* surfaces

$$\begin{pmatrix} x(u,v) \\ y(u,v) \\ z(u,v) \end{pmatrix} = \frac{\sum_{i=0}^{n} \sum_{j=0}^{m} w_{i,j} \boldsymbol{p}_{i,j} P_{i,n}(u) P_{j,m}(v)}{\sum_{i=0}^{n} \sum_{j=0}^{m} w_{i,j} P_{i,n}(u) P_{j,m}(v)},$$

where the $w_{i,j}$ are weights associated to each control point. The interest of rational surfaces is that they make the representation of particular surfaces (such as spherical parts) easier. The most commonly used surface representation in geometric modelling software falls into this category, since they make use of "Non-Uniform Rational B-Splines" or NURBS. We refer the reader to [102], [118] and [51] for in-depth studies regarding this type of surfaces. An example of such a surface is presented in Figure 3.3.



(a) Control mesh **(b)** Rendered surface



(c) Iso-u and iso-v curves

Figure 3.3: Example of a NURBS model

Although they provide high resolution models and are well-suited to a wide range of applications, piecewise polynomial surfaces are not ideal. Their major drawback is that a “real-world” surface, e.g. acquired from a laser scanner such as the statues scanned in the framework of the *Digital Michelangelo project* [87], are usually represented as polygonal meshes and turning them into piecewise-polynomial surfaces is a non-trivial operation. Moreover, at rendering time, the surface is tessellated into a polygonal mesh before being fed to the graphics hardware to be rendered, hence the interest of having powerful processing algorithms adapted to polygonal meshes.

3.2 Bivariate subdivision

A generalization of univariate subdivision to surface subdivision scheme is possible, but according to the differences between discrete functions and discrete surfaces presented in Section 3.1, may not be trivial. In this section, we will focus on a particular case of discrete surfaces, represented by a mapping between $\mathbb{Z}^2$ and $\mathbb{R}^3$. This class of discrete surfaces represents in fact regular rectangular meshes, which is naturally derived from the representation of the $\mathbb{Z}^2$ lattice. The generalization of subdivision schemes to data lying on $\mathbb{Z}^s$, $s > 1$ has been done simultaneously with the univariate case. Such an analysis can be found in [10] and [35]. Although the cases $s > 2$ are legitimate, we will limit ourselves to the case $s = 2$, since this is the only one of interest for surface subdivision.

The extension of subdivision schemes to data lying on $\mathbb{Z}^2$ will be presented in Section 3.2.1 and the analysis of such schemes will be detailed in Section 3.2.2.

3.2.1 Extending univariate schemes

Let us consider a rectangular mesh (or "quad-mesh"), i.e. a set of points $P_{k,l} \in \mathbb{R}^3$ (or $\mathbb{R}^n$, $n \geq 3$), with $(k,l) \in \mathbb{Z}^2$. The points $P_{k,l}$, $P_{k+1,l}$, $P_{k+1,l+1}$ and $P_{k+1,l}$ are connected together by edges and form a quadrilateral. As in the univariate case, the level to which each point belongs has to be indicated to avoid confusion. Thus, we will denote by $P^j_{k,l}$ the point of coordinates (k,l) at level j.

The analogous of the refinement equation (2.3) for such data is

$$P^{j+1}_a = \sum_{b\in\mathbb{Z}^2} \alpha^j_{a-nb} P^j_b, \ \forall a \in \mathbb{Z}^2. \tag{3.3}$$

The concepts defined in the framework of univariate subdivision, such as finite-support, stationarity and uniformity remain valid for bivariate subdivision. In this study, we will restrict ourselves to the binary case (i.e. $n = 2$ in equation (3.3)), although recently, Loop [90] proposed a ternary surface subdivision scheme. A particular case of binary schemes is to consider *tensor-product* schemes, i.e. to assume that each coefficient $\alpha_{k,l}$ ($\alpha^j_{l,k}$ in case of a non-stationary scheme) of the refinement mask satisfies

$$\alpha_{k,l} = \beta_k \beta_l,$$

where $\{\beta_k\}_{k\in\mathbb{Z}}$ is the mask of a univariate convergent scheme. Let us consider an example of such a scheme, using Chaikin's refinement rule as basis. We will represent the refinement rules as matrices, e.g.

$$P^{j+1}_{2k,2l} \leftarrow \begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix},$$

which is equivalent to

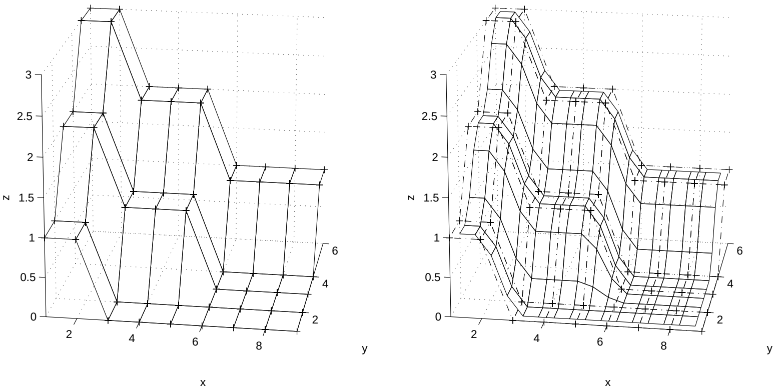
$$P^{j+1}_{2k,2l} = \alpha P^j_{k,l} + \beta P^j_{k+1,l} + \gamma P^j_{k,l+1} + \delta P^j_{k+1,l+1}.$$

Using the mask from Table 2.1, the bivariate refinement rules can be written

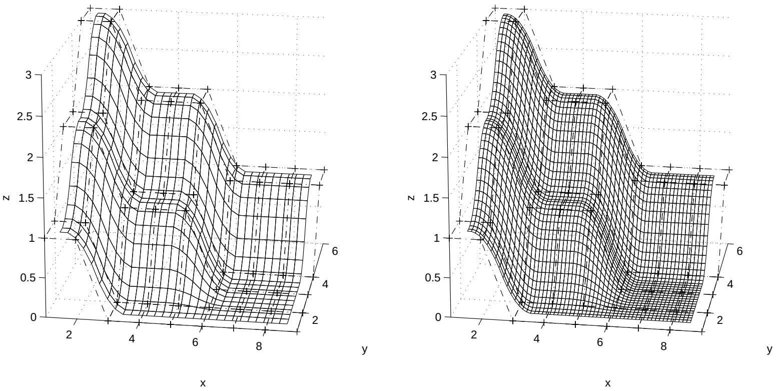
$$\begin{aligned} P^{j+1}_{2k,2l} &\leftarrow \tfrac{1}{16}\begin{pmatrix} 9 & 3 \\ 3 & 1 \end{pmatrix}, & P^{j+1}_{2k+1,2l} &\leftarrow \tfrac{1}{16}\begin{pmatrix} 3 & 9 \\ 1 & 3 \end{pmatrix}, \\ P^{j+1}_{2k,2l+1} &\leftarrow \tfrac{1}{16}\begin{pmatrix} 3 & 1 \\ 9 & 3 \end{pmatrix}, & P^{j+1}_{2k+1,2l+1} &\leftarrow \tfrac{1}{16}\begin{pmatrix} 1 & 3 \\ 3 & 9 \end{pmatrix}. \end{aligned} \tag{3.4}$$

This scheme is also known as the *Doo-Sabin* biquadratic scheme [32, 33]. The scheme is named after the authors of the original paper. The results obtained after performing two iterations of this scheme over a control mesh are shown in Figure 3.4.

Another related scheme was proposed at the same time (in the same journal issue) in [9], which is based on the tensor-product scheme formed by two cubic box-splines. This scheme became a popular scheme, referred to as the *Catmull-Clark* scheme (also from the names of the authors of the original paper). Nevertheless, this scheme is different from the previous one in that it performs



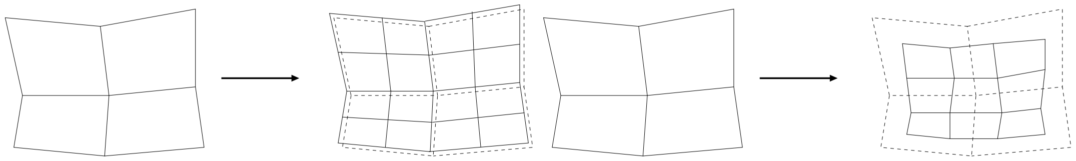
(a) Initial mesh **(b)** 1 level of subdivision



(c) 2 levels of subdivision **(d)** 3 levels of subdivision

Figure 3.4: Biquadratic spline (Doo-Sabin) tensor-product scheme based on Chaikin's rule

a *primal* refinement whereas the Doo-Sabin one is *dual.* A primal scheme will split *faces*, whereas the dual scheme will split *vertices.* In fact, the univariate refinements can also be distinguished between primal and dual. For instance, the quadratic spline (Chaikin) refinement is dual (splits the vertices and puts them part-way in the old edge), whereas the cubic spline scheme is primal (inserts a midpoint in the edge and modifies the old vertices). This concept is illustrated in Figure 3.5.



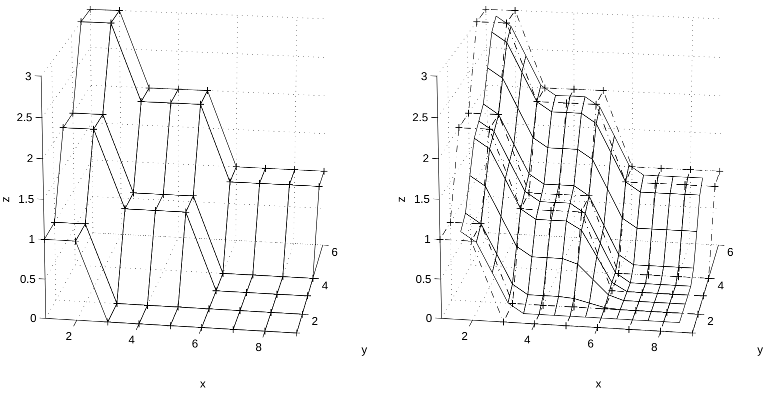
(a) Primal refinement **(b)** Dual refinement

Figure 3.5: Primal vs. dual refinement

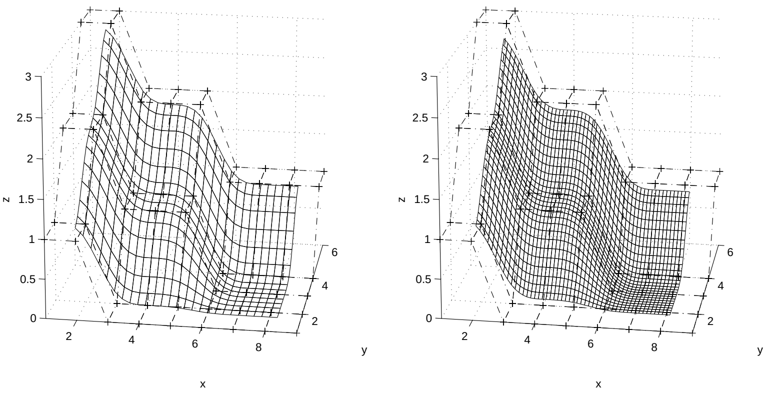
The insertion rules of this scheme are ($P^j_{k,l}$ corresponds to the central element of each matrix)

$$\begin{aligned}
P^{j+1}_{2k,2l} &\leftarrow \frac{1}{64}\begin{pmatrix}1 & 6 & 1\\ 6 & 36 & 6\\ 1 & 6 & 1\end{pmatrix}, & P^{j+1}_{2k+1,2l} &\leftarrow \frac{1}{16}\begin{pmatrix}0 & 1 & 1\\ 0 & 6 & 6\\ 0 & 1 & 1\end{pmatrix},\\
P^{j+1}_{2k,2l+1} &\leftarrow \frac{1}{16}\begin{pmatrix}0 & 0 & 0\\ 1 & 6 & 1\\ 1 & 6 & 1\end{pmatrix}, & P^{j+1}_{2k+1,2l+1} &\leftarrow \frac{1}{4}\begin{pmatrix}0 & 0 & 0\\ 0 & 1 & 1\\ 0 & 1 & 1\end{pmatrix}.
\end{aligned} \tag{3.5}$$

The rules used to compute $P^{j+1}_{2k+1,2l}$ and $P^{j+1}_{2k,2l+1}$ are termed "edge rules" in the original paper, since they compute the midpoints of edges. Similarly, the rule for $P^{j+1}_{2k,2l}$ is called a "vertex rule" and the rule for $P^{j+1}_{2k+1,2l+1}$ is called a "face rule". Figure 3.6 presents the effect of the Catmull-Clark subdivision over a control mesh.



(a) Initial mesh **(b)** 1 level of subdivision



(c) 2 levels of subdivision **(d)** 3 levels of subdivision

Figure 3.6: Bicubic spline tensor-product scheme (Catmull-Clark).

Another tensor-product scheme for quadrilateral meshes has been derived from the four-point scheme presented in Section 2.2 by Kobbelt in [74].

3.2.2 Analysis of bivariate schemes

Most of the criteria presented in Section 2.3 can be adapted to study bivariate subdivision. However, as we will show in this section, a non-negligible number of differences occur. Only the binary case will be studied.

As for univariate schemes, the convergence and regularity conditions are linked to the convergence of basis functions. In the case of a scheme being the tensor-product of two schemes $\mathcal{S}_a$ (for the x component) and $\mathcal{S}_b$ (for the y component), the expression of basis functions is separable

$$\phi_{\mathcal{S}_a \times \mathcal{S}_b}(x, y) = \phi_{\mathcal{S}_a}(x)\phi_{\mathcal{S}_b}(y). \tag{3.6}$$

Let us consider the general case of a bivariate scheme, having a (finite) mask $\{a_{i,j}\}_{(i,j)\in\mathbb{Z}^2}$. A bivariate polynomial $a(z_1, z_2) = \sum_{(i,j)\in\mathbb{Z}^2} a_{i,j} z_1^i z_2^j$ can be associated to this mask. A convergence condition close to the one in equation (2.19) holds

$$\sum_{p\in\mathbb{Z}^2} a_{l-2p} = 1, \; p \in \{(0,0), (0,1), (1,0), (1,1)\}.$$

However, unlike the univariate case, this condition only implies that

$$a(1,1) = 4, \; a(-1,1) = 0, \; a(1,-1) = 0, \; a(-1,-1) = 0, \tag{3.7}$$

but does not imply that the associated polynomial can be factorized similarly to equation (2.21).

In the case where a verifies $a(z_1, z_2) = (1 + z_1)(1 + z_2)b(z)$, the convergence of the scheme can be checked by verifying that the schemes associated to $(1 + z_1)b(z)$ and $(1 + z_2)b(z)$ are contractive. If we assume moreover that a can be factorized as

$$a(z_1, z_2) = (1 + z_1)^k (1 + z_2)^k b(z),$$

and that the schemes $a_{i,j}(z_1, z_2)$ defined by

$$a_{i,j}(z_1, z_2) = 2^{i+j} \frac{a(z_1, z_2)}{(1 + z_1)^i (1 + z_2)^j}, \; 0 \le i, j \le k,$$

are contractive, then the scheme produces $\mathcal{C}^k$ limit functions. Tensor-product schemes fall under this category of schemes, although in this particular case, according to equation (3.6), the smoothness of the limit function can be derived from the smoothness of the basis functions of the univariate schemes used to create the tensor-product scheme. As an example, the biquadratic scheme of equation (3.4) leads to $\mathcal{C}^1$ limit functions, whereas the bicubic scheme of equation (3.5) produces $\mathcal{C}^2$ limit functions.

The analysis in the general case is more difficult. We refer to [45, 54] for a summary of the analysis and to [10, 35] for more detailed presentations. An obvious consequence of conditions from equation (3.7) is that the two polynomials $(1 - z_1)a(z_1, z_2)$ and $(1 - z_2)a(z_1, z_2)$ are zeros at $(1, 1)$, $(1, -1)$, $(-1, 1)$ and $(-1, -1)$. Therefore, the following decomposition can be performed (not uniquely though)

$$\begin{aligned} (1 - z_1)a(z_1, z_2) &= b_{11}(z_1, z_2)(1 - z_1^2) + b_{12}(z_1, z_2)(1 - z_2^2), \\ (1 - z_2)a(z_1, z_2) &= b_{21}(z_1, z_2)(1 - z_1^2) + b_{22}(z_1, z_2)(1 - z_2^2). \end{aligned} \tag{3.8}$$

Let us denote the (bivariate) data at level j by $\{f_{k,l}^j\}_{(k,l)\in\mathbb{Z}^2}$. A relation, similar to the commutation property shown in Theorem 7, can be derived. In order to do so, we need to define the bivariate first order difference operator $\mathbf{\Delta}$ as

$$(\mathbf{\Delta} f)_{k,l} = \begin{pmatrix} f_{k,l} - f_{k-1,l} \\ f_{k,l} - f_{k,l-1} \end{pmatrix}.$$

As in Chapter 2, we denote by F^j the z-transform of f^j (in this case, this is a bivariate transform). A property of operator $\mathbf{\Delta}$ in the z domain is that

$$\mathbf{\Delta} F^j(z_1, z_2) = \begin{pmatrix} 1 - z_1 \\ 1 - z_2 \end{pmatrix} F^j(z_1, z_2). \tag{3.9}$$

Therefore, using the refinement relation $F^{j+1}(z_1, z_2) = a(z_1, z_2) F^j(z_1^2, z_2^2)$ and equation (3.8), we have

$$\mathbf{\Delta} F^{j+1}(z_1, z_2) = \begin{pmatrix} b_{11}(z_1, z_2) & b_{12}(z_1, z_2) \\ b_{21}(z_1, z_2) & b_{22}(z_1, z_2) \end{pmatrix} \begin{pmatrix} 1 - z_1^2 \\ 1 - z_2^2 \end{pmatrix} F^j(z_1^2, z_2^2). \tag{3.10}$$

Clearly, using equation (3.9), we have

$$\begin{pmatrix} 1 - z_1^2 \\ 1 - z_2^2 \end{pmatrix} F^j(z_1^2, z_2^2) = (\mathbf{\Delta} F^j)(z_1^2, z_2^2).$$

Therefore, equation (3.8) can be re-written as a commutation relation

$$\mathbf{\Delta} F^{j+1}(z_1, z_2) = \boldsymbol{B}(z_1, z_2)(\mathbf{\Delta} F^j)(z_1^2, z_2^2),$$

where $\boldsymbol{B}(z_1, z_2)$ is the z-transform of a matrix bivariate scheme,

$$\boldsymbol{B}(z_1, z_2) = \begin{pmatrix} b_{11}(z_1, z_2) & b_{12}(z_1, z_2) \\ b_{21}(z_1, z_2) & b_{22}(z_1, z_2) \end{pmatrix},$$

which corresponds to the matrix refinement rule

$$(\mathcal{S}_{\boldsymbol{B}} \boldsymbol{f})_l = \sum_{m \in \mathbb{Z}^2} \boldsymbol{B}_{l-2m} \boldsymbol{f}_m.$$

As in the univariate case, the scheme $\mathcal{S}_a$ is convergent if $\mathcal{S}_{\boldsymbol{B}}$ is contractive. The contractivity of this scheme can be verified by checking that $\|\mathcal{S}_{\boldsymbol{B}}^p\|_\infty$ is smaller than 1. The computation of the z-transform of $\mathcal{S}_{\boldsymbol{B}}^p$ is similar to the one shown in equation (2.25), since

$$\boldsymbol{B}^p(z_1, z_2) = \boldsymbol{B}(z_1, z_2) \boldsymbol{B}(z_1^2, z_2^2) \ldots \boldsymbol{B}(z_1^{2^{p-1}}, z_2^{2^{p-1}}),$$

and yields

$$\|\mathcal{S}_{\boldsymbol{B}}^p\|_\infty = \max_{l \in \{0 \ldots 2^p\}^2} \left\| \sum_{m \in \mathbb{Z}^2} |\boldsymbol{B}_{l-m2^p}^p| \right\|_\infty,$$

where $|A|$ denotes the matrix having elements that are the absolute value of the elements of A, and $\|A\|_\infty$ being the usual matrix infinite norm. Since the factorization of equation (3.8) is not unique, the contractivity condition is only a sufficient condition. The higher-order smoothness analysis is similar to the univariate case, except that the higher-order finite-difference give birth to matrices of matrices (...) of polynomials, and therefore makes the analysis process painful. Fortunately, the goal of subdivision is often to have a support that is as small as possible, leading to schemes that are usually not more than $\mathcal{C}^2$.

3.3 Subdivision of discrete surfaces

A further step in the generalization of subdivision was to consider not only bivariate (or n-variate) regular data, but data lying on more generic topologies. In fact, the Doo-Sabin and Catmull-Clark

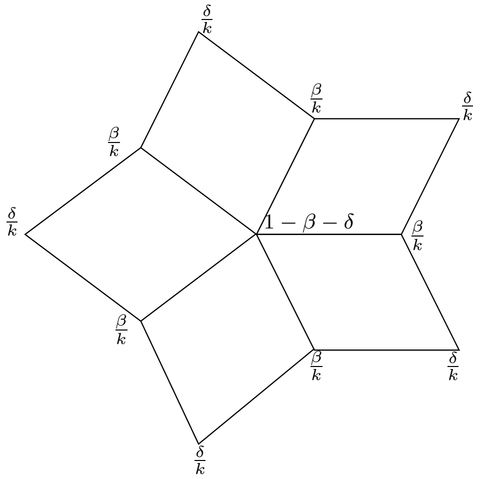


Figure 3.7: Vertex refinement rule for a point having k neighboring faces. The values of β and γ are $\beta = \frac{3}{2k}$ and $\delta = \frac{1}{4k}$.

schemes already addressed the cases of singularities in quad meshes. Specific rules were designed for vertices having k neighboring faces (Catmull-Clark, Kobbelt's tensor product of the four-point scheme) or k-gons (Doo-Sabin). Figure 3.7 illustrates the vertex rule of the Catmull-Clark scheme in the irregular case. In order to remain as generic as possible, and given that main concern of next chapter is triangular meshes and related subdivision algorithms, we will focus on this type of data. To describe such data, we will introduce the formalism of *topological nets* (also referred to as "simplicial complexes") in Section 3.3.1. The analysis techniques of linear subdivision schemes on such data will be detailed in Section 3.3.2.

3.3.1 Topological nets

The definitions developed in this section aim at generalizing the concepts of subdivision to more generic data, namely the discrete surfaces evoked in Section 3.1.2. Similar notations and concepts can be found in [97, 142] (in which they are termed "simplicial complexes") and [45, 74].

A topological net $\mathcal{N}$ is formed of three sets $\mathcal{N} = (V, E, F)$. V is a set of *vertices* from $\mathbb{R}^d$ ("classical" surface subdivision uses $d = 3$). Those vertices have topological relations described by the set of *edges* E and the set of *faces* F. An edge is a connection between two vertices, and a face is a cyclic list of vertices where all consecutive vertices are connected by an edge. Unlike the intuitive idea of a polygonal face in $\mathbb{R}^3$, this definition does not implies that the vertices belonging to a face are coplanar. Let us denote f the number of faces in F, v the number of vertices in V and by e the number of edges in E. Those quantities are linked through Euler's formula,

$$v - e + f = 2 - 2g, \tag{3.11}$$

where g is the *genus* of the net, i.e. roughly the number of "handles" in the net (e.g. $g = 0$ for a sphere and $g = 1$ for a torus). We refer to [94] for more precise definitions of this concept.

An edge is a *boundary edge* if it belongs to only one face. The vertices belonging to a boundary edge are termed *boundary vertices.* A few additional restrictions are necessary to have well-behaved topological nets, i.e. suitable to represent discrete surfaces:

1. a pair of vertices can share *at most* one edge, (i.e. no looping edges)

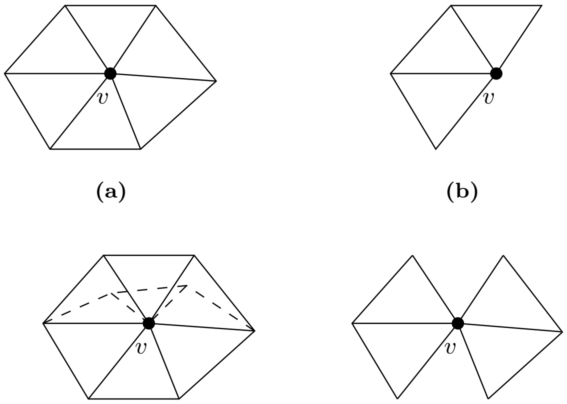
2. each vertex belongs to at least two distinct edges, (i.e. no dangling edge)

3. each face is formed by at least three distinct edges,

4. each edge can belong to at most two faces,

5. three boundary edges cannot share a vertex.

A net that violates one or more of these requirement is termed *non-manifold* (vs. *manifold* in the contrary). The qualification of "manifold" for discrete surfaces represented by such topological nets is more restrictive that the definition given by differential geometry. Examples of manifold and non-manifold cases are presented in Figure 3.8. Figures 3.8(a) and 3.8(b) present repectively a manifold vertex and a manifold boundary vertex. The example of Figure 3.8(c) contains non-manifold edges that violate condition 4 of the above list, while the example in Figure 3.8(d) contains a non-manifold vertex, and violates condition 5.

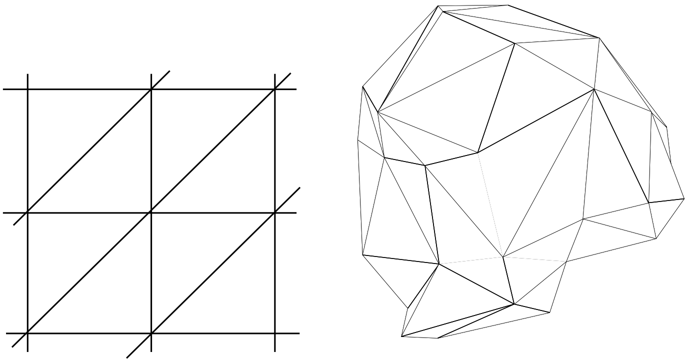


(a) **(b)**

(c) **(d)**

Figure 3.8: Illustration of manifold vs. non-manifold

A net having no boundary edges/vertices is termed *closed* (vs. *open* in the contrary). The number of edges that have a common vertex v_i is termed the *valence* of v_i. The number of edges forming a face f_k is also termed "valence" of f_k. The examples described in Section 3.2 used data lying on $\mathbb{Z}^2$, which can be considered as a quadrilateral net: each vertex and each face has a valence equal to 4. Another case evoked in Section 3.1 is triangular mesh, i.e. a net where all faces have a valence equal to 3 (if the net is closed). In this case, there is no *a priori* knowledge on the valence of the vertices. Still, a distinction is made between *regular* and *irregular* triangular nets: a triangular net is regular if all vertices are of valence 6, which corresponds to the intuitive triangulation of the $\mathbb{Z}^2$ quadrilateral net. An example of regular and irregular nets is presented in Figure 3.9.



(a) Regular triangulation of $\mathbb{Z}^2$ **(b)** Irregular triangular net

Figure 3.9: Regular vs. irregular triangular nets

An analogous of linear subdivision presented for univariate and bivariate data can be adapted to topological nets. A subdivision operator turns a net $\mathcal{N} = (V, E, F)$ into a refined net $\mathcal{N}' = (V', E', F')$. The vertices from V' are obtained by linear combinations of vertices from V. Each vertex v' from V' is associated with an element t of $\mathcal{N}$ (i.e. associated with a vertex, edge or face) and the weights used in the linear combination only depend on the topological relations between vertices form V and t. Depending on the type of the net, different refinements are available.

When no *a priori* assumption is made on the topological structure of the net, two main refinements are available: the *primal* and *dual* refinements, briefly introduced in Section 3.2.1 and illustrated by Figure 3.5. We refer the reader to [119] for an intuitive illustration of all primal/dual subdivision schemes on both arbitrary and triangular nets.

The primal refinement (e.g. Catmull-Clark scheme presented previously) splits each face of valence k into k quadrilateral faces. At each refinement steps, the vertices can be classified into three categories:

- vertices associated with a face (f-vertices), roughly the “middle” of the face
- vertices associated with a edge (e-vertices), which can be seen as the midpoint of the edge to which they are related
- vertices associated with a vertex (v-vertices), which are the updated positions of vertices from V, for instance using the Catmull-Clark rule shown in Figure 3.7.

The different types of vertices are illustrated by Figure 3.10. Such a refinement generates semi-regular quadrilateral nets, since all new e-vertices have a valence equal to 4. The f-vertices have a valence equal to the valence of the face they refer to in F, and v-vertices have a valence equal to the one of the vertex they refer to in V. After several applications of this refinement, only isolated vertices will have a valence not equal to 4 (*irregular* vertices), hence the *semi-regular* designation.

The dual refinement (e.g. the Doo-Sabin scheme) acts differently, in that each new vertex in V' is associated to a face $f \in F$ and to a vertex $v \in V$, such that v is a vertex of f. As suggested previously in Figure 3.5, a dual refinement splits a vertex of valence k into k new vertices (and therefore give birth to a face of valence k in the next level). The faces in the refined net therefore correspond to edges and vertices in the original net. This refinement is illustrated in Figure 3.11.

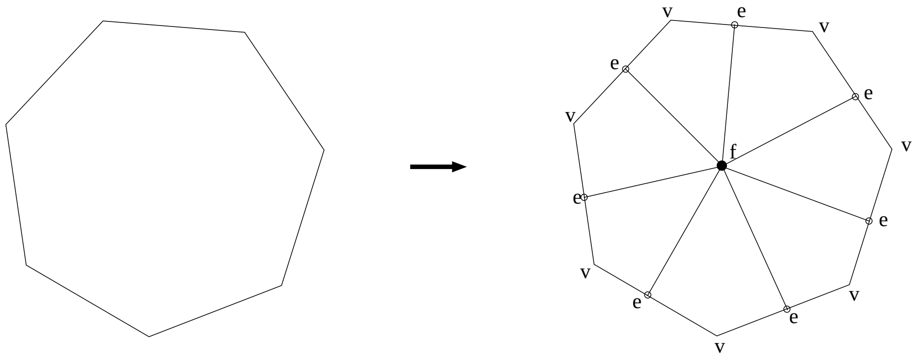


Figure 3.10: Primal refinement of a k-sided face. Letters indicate the type of each newly computed vertex

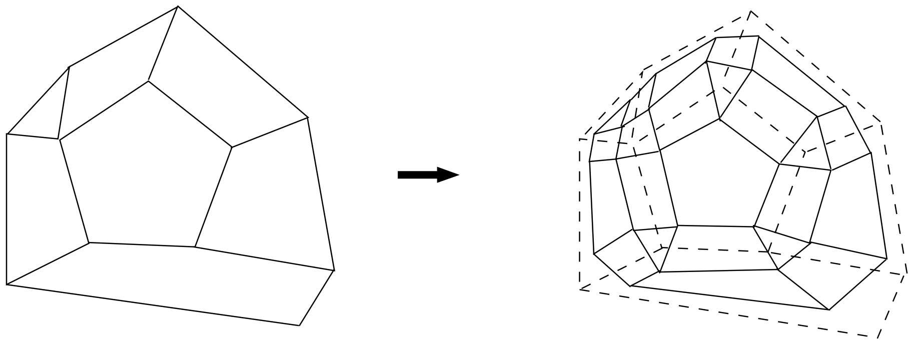


Figure 3.11: Dual refinement of a net

As mentioned above, a particular case of topological net that is of interest is the triangular net. In this case, a refined net will be formed by only two types of vertices, namely v-vertices (i.e. the updated positions of the original vertices from V) and e-vertices, related to an edge in the initial net (acting as midpoint for the "parent" edge), therefore splitting each triangle into four new triangles, as shown in Figure 3.12.

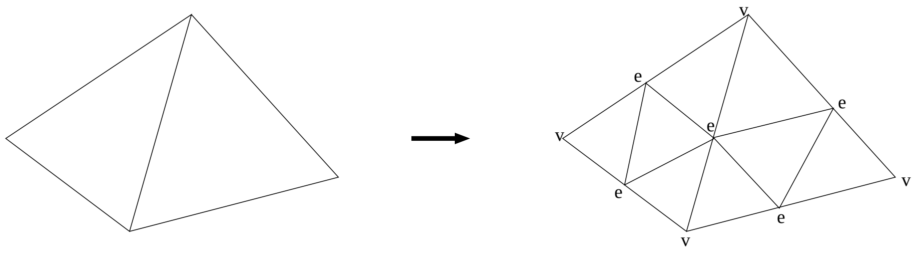


Figure 3.12: Refinement of a triangular net. Letters indicate the type of each vertex.

In this case, the number of triangles in the net is multiplied by four at each refinement step. All the e-vertices inserted are of valence 6, while the v-vertices keep the valence of their respective parent vertex. In the case of an interpolating refinement, the v-vertices are "copies" from their parent vertices. After several refinement steps, the irregular vertices will be isolated and as for the

Catmull-Clark scheme, the resulting net is called *semi-regular.*

Another approach for triangular net subdivision was proposed by Kobbelt in [76], called $\sqrt{3}$-subdivision. Instead of inserting a new e-vertex for each edge (i.e. 3 per triangle), only one f-vertex (midpoint of the triangle) is inserted at each step. Each triangle is split into three new triangles, and the edges in the original net are flipped, in order to have a semi-regular mesh. This technique is illustrated by Figure 3.13. An interpolating scheme using this refinement technique has been proposed in [83].

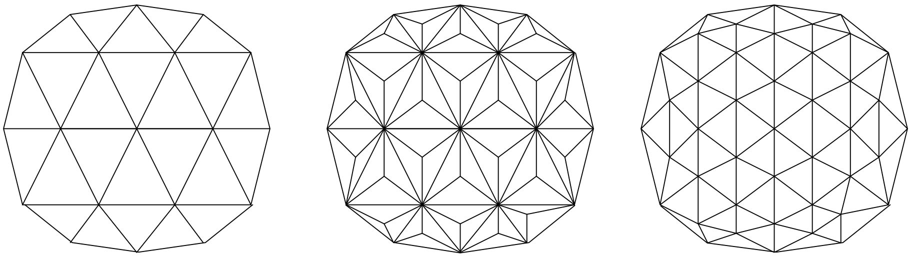


Figure 3.13: Illustration of the two steps of the $\sqrt{3}$-subdivision. Midpoints are inserted in the "middle" of each face (central picture) and then original edges are flipped (right picture)

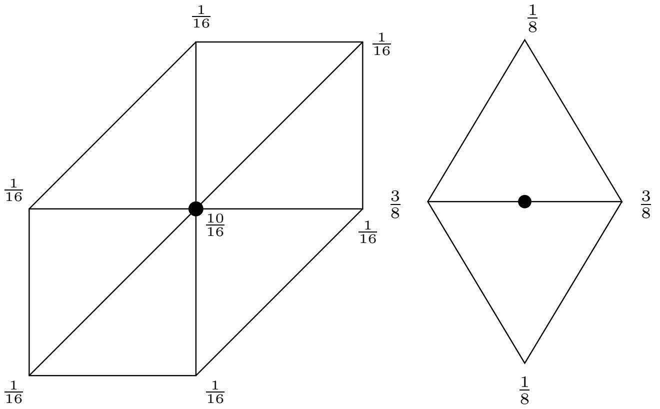
A famous example of triangular subdivision has been proposed by Charles Loop in [88]. The subdivision method he proposed was based on a quadratic box-spline, similar to the one presented in Section 2.2, except that it had *three* directions to handle regular triangulations (like the one of $\mathbb{Z}^2$ shown in Figure 3.9(a)). The associated polynomial for Loop's scheme is

$$a(z_1, z_2) = \frac{1}{16} z_1^{-2} z_2^{-2} (1 + z_1)^2 (1 + z2)^2 (1 + z_1 z_2)^2,$$

the diagonal direction being materialized by the factor $1 + z_1 z_2$. The refinement rules derived from this mask are

$$\begin{aligned}
P_{2k,2l}^{j+1} &= \frac{5}{8} P_{k,l}^{j} + \frac{1}{16}\left(P_{k-1,l-1}^{j} + P_{k-1,l}^{j} + P_{k,l-1}^{j} + P_{k+1,l+1}^{j} + P_{k+1,l}^{j} + P_{k,l+1}^{j}\right),\\
P_{2k+1,2l}^{j+1} &= \frac{3}{8}\left(P_{k,l}^{j} + P_{k+1,l}^{j}\right) + \frac{1}{8}\left(P_{k,l-1}^{j} + P_{k+1,l+1}^{j}\right),\\
P_{2k,2l+1}^{j+1} &= \frac{3}{8}\left(P_{k,l}^{j} + P_{k,l+1}^{j}\right) + \frac{1}{8}\left(P_{k-1,l}^{j} + P_{k+1,l+1}^{j}\right),\\
P_{2k+1,2l+1}^{j+1} &= \frac{3}{8}\left(P_{k,l}^{j} + P_{k+1,l+1}^{j}\right) + \frac{1}{8}\left(P_{k+1,l}^{j} + P_{k,l+1}^{j}\right).
\end{aligned} \tag{3.12}$$

These refinement rules can be classified according to the categories illustrated by Figure 3.12. $P_{2k,2l}^{j+1}$ is clearly an update of the position of $P_{k,l}^{j}$, i.e. a v-vertex, whereas the others are midpoints for the edges of the triangulation, i.e. e-vertices. Figure 3.14 shows the stencils associated to the refinement rules of equation (3.12), for a closed regular triangular net. As implied by the criteria developed Section 3.2.2, the Loop scheme generates $\mathcal{C}^2$ functions on regular triangulations. The scheme can be adapted to handle irregular vertices, by modifying the weights of the v-vertex rule. As the derivation of these rules requires further analysis, this will be developed in Section 3.3.2.



(a) v-vertex rule **(b)** e-vertex rule

Figure 3.14: Loop subdivision rules for regular closed triangular nets. New vertices (or new position for v-vertices) are materialized by the black dot.

Another widely used subdivision technique for triangular nets is the "Butterfly" scheme. As for the Loop scheme, this method has been derived by adapting a univariate rule to regular triangular nets. In this case, the binary four-point scheme studied in Chapter 2 was used as basis as detailed in [46]. Since the univariate four-point scheme interpolates data, the resulting scheme for triangulations will also be interpolating, i.e. the v-vertex rule is $P^{j+1}_{2k,2l} = P^j_{k,l}$. Unlike Loop's scheme, the e-vertex rules are not simply derived from a simple tensor-product, but from a convergence condition which we will developed in Section 3.3.2. Those rules are

$$\begin{aligned}
P^{j+1}_{2k+1,2l} &= \frac{1}{2}\Big(P^j_{k,l} + P^j_{k+1,l}\Big) + 2w\Big(P^j_{k,l-1} + P^j_{k+1,l+1}\Big) \\
&\quad - w\Big(P^j_{k-1,l-1} + P^j_{k+1,l-1} + P^j_{k,l+1} + P^j_{k+2,l+1}\Big), \\
P^{j+1}_{2k,2l+1} &= \frac{1}{2}\Big(P^j_{k,l} + P^j_{k,l+1}\Big) + 2w\Big(P^j_{k-1,l} + P^j_{k+1,l+1}\Big) \\
&\quad - w\Big(P^j_{k-1,l-1} + P^j_{k-1,l+1} + P^j_{k+1,l} + P^j_{k+1,l+2}\Big), \\
P^{j+1}_{2k+1,2l+1} &= \frac{1}{2}\Big(P^j_{k,l} + P^j_{k+1,l+1}\Big) + 2w\Big(P^j_{k+1,l} + P^j_{k,l+1}\Big) \\
&\quad - w\Big(P^j_{k,l-1} + P^j_{k-1,l} + P^j_{k+2,l+1} + P^j_{k+1,l+2}\Big),
\end{aligned} \tag{3.13}$$

and can be summarized by Figure 3.15, since they are symmetrical with respect to the edges they split. The shape of the mask is the origin of the name "Butterfly" given to the scheme. The subdivision rule from equation (3.13) is valid only for *regular edges*, i.e. edges connecting two regular vertices. An extension of this method to handle irregular vertices has been developed and will be detailed in the next section. The polynomial associated to the regular Butterfly scheme can be factored as

$$a(z_1, z_2) = \frac{1}{2z_1z_2}(1 + z_1)(1 + z_2)(1 + z_1z_2)(1 - wb(z_1, z_2)),$$

which, using a generalization [49] of the analysis presented in Section 3.2.2, implies that the convergence of the scheme depends on the contractivity of the schemes associated to the following polynomials

$$
\begin{aligned}
d_1(z_1, z_2) &= \frac{1}{z_1 z_2}(1 + z_1)(1 - wb(z_1, z_2)),\\
d_2(z_1, z_2) &= \frac{1}{z_1 z_2}(1 + z_2)(1 - wb(z_1, z_2)),\\
d_3(z_1, z_2) &= \frac{1}{z_1 z_2}(1 + z_1 z_2)(1 - wb(z_1, z_2)).
\end{aligned}
$$

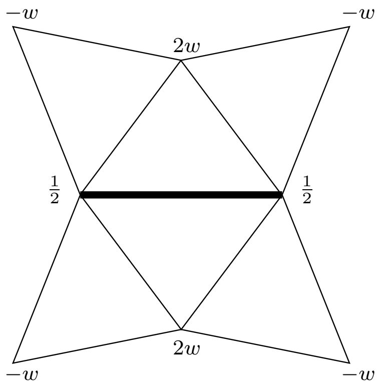


Figure 3.15: Butterfly mask for a regular edge (in bold).

As explained in [39], the Butterfly scheme is $\mathcal{C}^1$ for $0 < w < 0.096$ at least. It is worth pointing that $w = \frac{1}{16}$, for which the scheme reproduces cubic polynomials on regular triangulations is within this range of convergence.

As a side note, there are a few schemes that are dedicated to subdividing hexagonal nets. We refer the reader for instance to [48] or [13] for an example of such a technique. In fact, hexagonal nets can be seen as *dual nets* of regular triangular nets (i.e. obtained by connecting the centers of neighboring triangles by edges). As a consequence, the triangle splitting techniques illustrated by Figure 3.12 or Figure 3.13 are often termed *primal* whereas hexagonal subdivision is considered as *dual* subdivision for triangular nets.

3.3.2 Analysis of surface subdivision schemes

As suggested in Sections 3.2 and 3.3.1, the convergence of surface subdivision scheme for a *regular* vertex can be inferred from a generalization of univariate case. Here, the notion of “regular vertex” depends on the type of topological net and subdivision scheme under consideration (e.g. vertex of valence 6 for triangular subdivision, vertex of valence 4 for quadrilateral subdivision...). Even at the early times of surface subdivision, Doo and Sabin [32], as stated in the title of their paper, already proposed what became the basis of the analysis tools of subdivision surfaces around extraordinary points.

The analysis techniques of linear surface subdivision schemes relies on the properties of the eigenvalues of a *subdivision matrix* which represents the subdivision rules around the extraordinary vertex. This was originally proposed in [32], and further developed in many sequels. Ball and Storry [4, 5] did the pioneer work in checking the relationship between the eigenvalues of the subdivision matrix around an extraordinary vertex and $\mathcal{C}^1$-continuity of the limit surface. Their work proved a relation between the eigenvalues of the subdivision matrix and the continuity of the normal vectors to the surface (or *tangent-plane continuity*), which is weaker than $\mathcal{C}^1$-continuity. Using such an analysis, Zorin, Schröder and Sweldens generalized the Butterfly scheme to generate $\mathcal{C}^1$-surfaces at irregular vertices [145]. The works from Ball and Storry were extended by Reif [109, 110], in which he carried a more rigorous analysis and gave sufficient conditions to achieve $\mathcal{C}^1$-regularity, in the differential geometry meaning (i.e. the existence of a smooth parameterization as explained in Section 3.1), of the limit surface. The proof that this condition was also necessary was given in [101]. This method has been used to analyze piecewise smooth schemes in [120]. The criteria given by Reif were valid for an arbitrary large but finite number of valences, but were extended for any valence by Zorin in [143]. The extension of these criteria to derive $\mathcal{C}^k$-continuity criteria were mostly studied by Prautzsch [103, 104, 105] and Zorin [141, 142, 144]. Developing similar techniques for triangular nets, Umlauf [131] carried the analysis of Loop's scheme. Finally Schröder and Reif showed in [111] that Loop and Catmull-Clark schemes produced surfaces which principal curvatures were square-integrable. In this section, we will summarize (briefly) the main results regarding the analysis of linear surface subdivision schemes.

The definitions convergence and regularity of surface subdivision schemes are similar to the ones presented in Chapter 2. Using the formalism of topological nets presented in this section, their adaption is rather straightforward. Let us consider a sequence of topological nets $\{\mathcal{N}^j = (V^j, E^j, F^j)\}_{j\in\mathbb{N}}$, obtained through successive refinements of the initial net $\mathcal{N}^0$.

As stated in the introduction, the analysis of surface subdivision schemes is based on the subdivision matrix associated to the scheme. Given a vertex $n+1$ vertices v_p^j $\{v_p^j\}_{p\in\{0\ldots n\}}$, the subdivision step can be represented by a square $n+1 \times n+1$ matrix S which describes the subdivision rules

$$\begin{pmatrix} v_0^{j+1} \\ v_1^{j+1} \\ \vdots \\ v_k^{j+1} \end{pmatrix} = S \begin{pmatrix} v_0^{j} \\ v_1^{j} \\ \vdots \\ v_n^{j} \end{pmatrix}. \tag{3.14}$$

For instance the vertices v_p^j consist in one irregular vertex v_0^j and n "neighbors", in the widest meaning. For instance, these could be vertices sharing an edge with the irregular vertex (also called *1-ring* of v_0^j), or further vertices, for instance sharing an edge with the vertices from the 1-ring of v_0^j (or *2-ring* of v_0^j). An example around an irregular vertex is presented in Figure 3.16. In this figure, only the 1-ring of the irregular vertex is involved.

An example of subdivision matrix as defined in equation (3.14), for the Loop scheme over the configuration ($n = 7$) presented in Figure 3.16 can be written as follows,

$$S = \frac{1}{8}\begin{pmatrix} \alpha & \beta & \beta & \beta & \beta & \beta & \beta & \beta \\ 3 & 3 & 1 & 0 & 0 & 0 & 0 & 1 \\ 3 & 1 & 3 & 1 & 0 & 0 & 0 & 0 \\ 3 & 0 & 1 & 3 & 1 & 0 & 0 & 0 \\ 3 & 0 & 0 & 1 & 3 & 1 & 0 & 0 \\ 3 & 0 & 0 & 0 & 1 & 3 & 1 & 0 \\ 3 & 0 & 0 & 0 & 0 & 1 & 3 & 1 \\ 3 & 1 & 0 & 0 & 0 & 0 & 1 & 3 \end{pmatrix}. \tag{3.15}$$

The first line of the matrix represents the v-rule for the irregular vertex, whereas the rest summarizes the e-rules for the vertices of the 1-ring of v_0^j. However, the scheme cannot be directly studied using such a matrix. In fact, depending on the scheme, the order of the ring surrounding the extraordinary point to be considered has to be greater. If we suppose there is only one extraordinary point, the number of rings to consider has to be such that there is at least one complete ring of fully regular points (i.e. not sharing an edge with the irregular vertex). For the Loop scheme, 3 rings have to be considered. Using an appropriate numbering of the vertices, the subdivision matrix around an extraordinary vertex can be written as a block-circulant matrix, and therefore can be block-diagonalized by applying a Fourier transform.

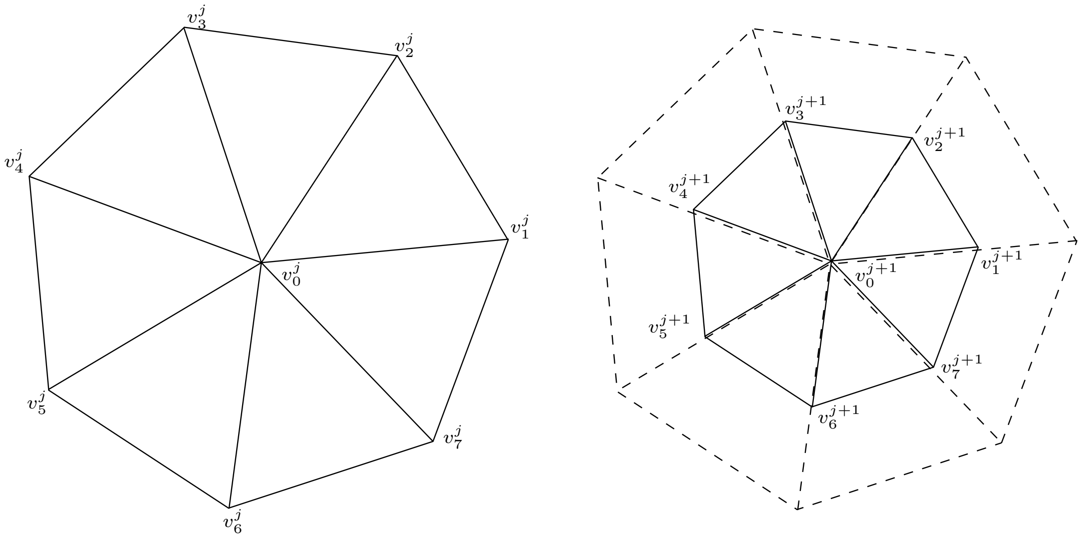


Figure 3.16: Vertex labelling of the 1-ring around an extraordinary vertex v_0^j

The conditions on the eigenvalues initially formulated by Ball and Storry and refined by Reif and Zorin, are now summarized. Let us denote by $\lambda_0, \lambda_1, \ldots, \lambda_n$ the eigenvalues of S such that

$$|\lambda_0| \geq |\lambda_1| \geq \cdots \geq |\lambda_n|.$$

In order for the subdivision scheme to be affine invariant, the rows of the subdivision matrix all sum up to 1 (which implies that $\beta = \frac{1-\alpha}{n}$ in equation (3.15)), which also yields that 1 is always an eigenvalue of the subdivision matrix. As stated by Reif, the subdivision scheme converges if $|\lambda_1| < \lambda_0 = 1$. The study of $\mathcal{C}^1$-convergence is less trivial. The conditions on the eigenvalues of the matrix for the scheme to be $\mathcal{C}^1$ remain still simple. The initial condition on the eigenvalues derived by Reif in [110] were that it was necessary to have $\lambda_1 = \lambda_2$ and $|\lambda_3| < |\lambda_1| < 1$ in order to have a regular limit surface. In a sequel [109], Reif showed that it was only necessary to have

$$|\lambda_3| < |\lambda_2| \leq |\lambda_1| < 1 = \lambda_0. \tag{3.16}$$

While the conditions on the eigenvalues are relatively easy to check, they are not sufficient to ensure the regularity of the limit surface. One has also to check that the limit surface can be parameterized appropriately. In order to check this, Reif introduced the notion of *characteristic map* around an extraordinary vertex. In order to define this mapping, we need to consider the neighborhood around an extraordinary vertex, represented by a ring of parametric patches. At each refinement step, a linear transformation maps the set of patches $\mathcal{P}^j = \bigcup \mathcal{P}_l^j$ to the refined set of patches $\mathcal{P}^{j+1}$ (cf. Figure 3.17 for an example of such a configuration). The subdivision scheme converges if the closure of the sequence $\{\mathcal{P}^j\}$ has no gaps.

Each set of patches $\mathcal{P}^j$ is influenced by a set of control points $P^j = \{P_r^j\}_{r \in I}$, and each patch $\mathcal{P}_l^j$ is influenced by a subset of P^j, $\{P_m^j\}_{m \in I_l}$. If we denote by ϕ the limit function of the subdivision scheme under consideration on a regular grid, each parametric patch is a set of functions

$$\mathcal{P}_l^j = \left\{ p_l^j(u,v) = \sum_{m \in I_l} P_m^j \phi(2^j u - d_m, 2^j v - e_m),\ (u,v) \in \Omega \right\}, \tag{3.17}$$

where Ω is the parameter domain, e.g. $\Omega = \{(u,v),\ 0 \leq u,v \leq 1\}$ for a quadrilateral net, and $\Omega = \{(u,v),\ u,v \geq 0 \text{ and } u+v \leq 1\}$ for a triangular one. In order to ensure the regularity of the parameterization, the subdivision process has to generate $\mathcal{C}^1$ functions on regular grids, i.e. ϕ must be $\mathcal{C}^1$.

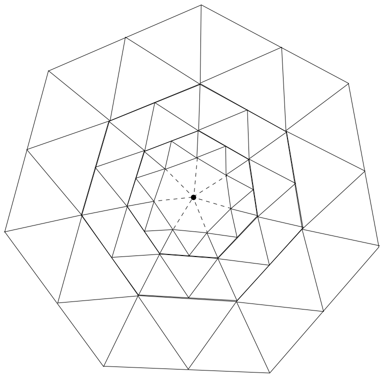


Figure 3.17: 3 successive rings of patches around an extraordinary vertex (materialized by the dot). Each ring is twice smaller than the previous one

Definition 12. *Let us consider a subdivision matrix of size $N \times N$ associated to a scheme. Let us denote by v_1 and v_2 the eigenvectors associated to λ_1 and λ_2. In case where $\lambda_1 = \lambda_2$, the generalized eigenvectors can be used to define a Jordan basis of the associated subspace. A set of control points P^0 can be associated to these eigenvectors is defined by*

$$P^0 = \left\{ \begin{pmatrix} v_{1,j} \\ v_{2,j} \\ 0 \end{pmatrix} \right\},\ j \in \{1, \ldots, N\},$$

where $v_{i,j}$ denotes the j-th coordinate of v_i. The ***characteristic map*** *associated to the scheme is the set $\mathcal{P}^0 = \{p_l^0\}$, as defined in equation (3.17), associated to this initial data set P^0.*

The following theorem from Reif summarizes the conditions to achieve regularity of the limit surface.

Theorem 15. *Assuming that the subdivision matrix has eigenvalues meeting the condition of equation (3.16), and that the patches only overlap at their boundaries, i.e. $\bigcap_l \overset{\circ}{\mathcal{P}_l^0} = \emptyset$, the scheme is tangent-plane continuous (i.e. the normals to the surface are continuous) is the characteristic map in regular, i.e. if the Jacobian of the characteristic map in never equal to 0 for all (u,v). If moreover the characteristic map is injective, the limit surface is $\mathcal{C}^1$ for almost every initial data (i.e. everywhere except a nowhere dense set of configurations).*

While this theorem provides a rigorous way of checking the regularity of the limit surface around an extraordinary point, its application is not straightforward. More recently, Zorin [143] has exploited some conditions basically equivalent to the above theorem, but that can be implemented to check numerically the convergence of a given subdivision technique, mostly based on linear approximations of the characteristic map (which must be of constant sign) and on the winding number of a curve surrounding the extraordinary vertex.

This analysis has been extended to derive conditions for $\mathcal{C}^k$-continuity of the limit surface, in [103, 104] and [141]. Let us denote by v_1 and v_2 the (generalized) eigenvectors associated to λ_1 and λ_2. Although the conditions are very similar, a distinction is made when λ_1 and λ_2 are real eigenvalues and when $\lambda_1 = \bar{\lambda_2}$.

Theorem 16. *If λ_1 and λ_2 are real, the associated scheme produces $\mathcal{C}^k$ limit surfaces if*

- *The characteristic map (v_1, v_2) is regular and invertible*
- *For all eigenvalues λ such that $|\lambda_1| \geq |\lambda| \geq |\lambda_2|^k$ have equal algebraic and geometric orders and their associated eigenspace is*

$$\mathit{span}\{v_1^\alpha v_2^\beta,\ \lambda = \lambda_1^\alpha \lambda_2^\beta, (\alpha, \beta) \in \mathbb{N}^2\}, \tag{3.18}$$

- *All other eigenvalues are smaller in module than $|\lambda_2|^k$.*

(which implies that the eigenvalues are either smaller than $|\lambda_2|^k$ or can be written as $\lambda_1^\alpha \lambda_2^\beta$)

If the eigenvalues are complex conjugates, the condition of equation (3.18) *remains identical, and the characteristic map is given by* $(\mathfrak{Re}\, v_1, \mathfrak{Im}\, v_1)$.

The second condition of this theorem roughly means that the functions composing the patches (cf. equation (3.17)) surrounding the extraordinary point are homogeneous polynomials of degree k. However, this theorem does not provide a way of constructing a $\mathcal{C}^k$ scheme for $k > 1$. Even the problem of having $\mathcal{C}^2$ limit surfaces is only partially solved. The most recent attempts in [106, 107] and [89] extended subdivision techniques to obtain surfaces with bounded curvature.

The conditions for $\mathcal{C}^1$ continuity on the subdivision matrix of Loop scheme around an extraordinary vertex having n neighbors lead to study a block-circulant matrix A of size $7n \times 7n$ (provided the vertices surrounding the irregular points have been labelled in an appropriate order),

$$A = \begin{pmatrix} A_0 & A_1 & \cdots & A_{n-1} \\ A_{n-1} & A_0 & \cdots & A_{n-2} \\ \vdots & & \ddots & \vdots \\ A_1 & \cdots & A_{n-1} & A_0 \end{pmatrix},$$

where each block A_p is a 7×7 matrix. Applying the discrete Fourier transform on this matrix yields a unitary similar block-diagonal matrix. We will denote by $\hat{A}_p$, $p = 0, \ldots, n$ the diagonal blocks of the transformed matrix. The eigenvalues of this matrix are (cf. [131]) $\lambda_0 = 1$, $\lambda_\alpha = \alpha - \frac{3}{8}$, $\lambda_p = \frac{3}{8} + \frac{1}{4}\cos\frac{2p\pi}{n}$ for $p = 1, \ldots, n-1$, $\frac{1}{8}$ and $\frac{1}{16}$ each with order n, and finally 0 with order $4n - 1$. Moreover, we have $\lambda_{n-m} = \lambda_m$ for $m = 1, \ldots, \lfloor \frac{n}{2} \rfloor$ and $\lambda_1 > \lambda_m$ for $m = 2, \ldots, \lfloor \frac{n}{2} \rfloor$. In addition, we have $\hat{A}_1$ and $\hat{A}_{n-1}$ that are adjoints matrices, which implies that λ_1 has a geometric order equal to 2. The condition for $\mathcal{C}^1$ regularity is then $|\lambda_\alpha| < \lambda_1$ which finally leads to

$$\alpha \in \left(-\frac{1}{4}\cos\frac{2\pi}{n},\ \frac{3}{4} + \frac{1}{4}\cos\frac{2\pi}{n} \right).$$

Loop's choice for α was $\left(\frac{3}{8} + \frac{1}{4}\cos\frac{2\pi}{n}\right)^2 + \frac{3}{8}$, while Warren [135, 136] proposed $\alpha = \frac{5}{8}$ (except for $n = 3$ where $\alpha = \frac{7}{16}$), which produces visually equivalent results to Loop's choice.

In this chapter we have reviewed the linear surface subdivision techniques. They rely mostly on generalizations of univariate technique for the regular setting, but require more care when dealing with irregularities. The rules to achieve a smooth limit surface around an extraordinary point are strongly dependent on its valence. Unlike univariate schemes, regularity analysis is usually not carried beyond $\mathcal{C}^1$ in the irregular case, or considers only boundedness of the curvature of the surface.

Non-Linear Subdivision

4

As shown in Chapters 2 and 3, linear subdivision has been widely studied, although several open questions still persist, especially in the case of discrete surface subdivision. A relatively natural generalization of linear subdivision, although not straightforward is to consider a relation more generic (such as equation (2.1)) than the linear refinement equation 2.3, not being necessarily formed by a linear combination but by "any" operator, possibly depending on the data itself. Some steps toward fully non-linear subdivision have been done by considering linear univariate schemes perturbed by a non-linear term. The aim of such methods was to design schemes having specific properties that could not be obtained with purely linear schemes, for instance convexity and/or monotonicity preservation. This chapter is organized as follows: Section 4.1 will present generalities regarding non-linear subdivision and describe the principal existing non-linear subdivision schemes and analysis methods. Section 4.2 will present a original fully non-linear technique that uses a local coordinate system. This technique will be partially analyzed in Section 4.3.

4.1 Generic non-linear subdivision

In this section we generalize some definitions of Section 2.1.2 to describe and analyze non-linear subdivision operators. A few existing non-linear subdivision techniques will be presented.

4.1.1 Previous work

Whereas many studies concerning linear and stationary subdivision operators can be found, works concerning non-linear and/or non-stationary/non-uniform cases remain relatively sparse. The main results regarding the linear case have been shown in chapters 2 and 3. Several works concerning particular non-linear schemes have also been performed. In [52], a particular case of schemes made of a linear part and a non-linear part, in this case the harmonic mean of a linear combination of sample, is presented. This scheme was aiming at preserving the convexity of the original data. this concept was developed further by Kuijt and van Damme in [78] and [40, 79, 80, 81, 82]to design convexity and monotonicity preserving schemes, since linear subdivision schemes cannot in general guarantee the preservation of these properties. Another approach has been proposed, based on the essentially non-oscillatory (ENO) methods in [15]. The scheme proposed in this paper resembles the linear operator, but with coefficients depending on the data, and the aim was to capture the singularities (i.e. edges) in images. This paper however provides a few interesting criteria regarding data-dependent schemes, that extend the ones derived by Kuijt and van Damme into a more generic setting. The results of this paper have been used as basis by Oswald in [99], where he extends further the results of [15].

While most of the stationary linear techniques have been extended to discrete triangulated surfaces, non-linear surface subdivision schemes are not common. A method of convexity-preserving surface subdivision, based on geometric considerations has been proposed in [47]. Another interpolating approach has been described in [70], which resembles by some aspects the proposed scheme described in Section 4.2. However, according to the author, the results achieved in terms of smoothness of the limit surface are not really satisfactory and require a smoothing step.

4.1.2 Analysis of non-linear schemes

The analysis of such schemes can be done using adaptations of the criteria seen in Section 2.3. Clearly, the Laurent polynomials framework cannot be used directly. In this section, we present the main results regarding a particular class of non-linear subdivision schemes derived by Cohen et al. in [15] and the extension done by Oswald in [99].

Let us focus on the univariate binary case (a generalization to the n-ary case is rather straightforward). We consider a refinement operation as in equation (2.1), i.e.

$$(X_{j+1}, F_{j+1}) = \mathcal{S}_j(X_j, F_j),$$

where, as in Chapter 2, X_j and F_j are isomorphic to the canonical grid $\{k2^{-j}\}$, and the values in F_j are points in $\mathbb{R}^d$. The qualifiers introduced for linear schemes are still valid although slightly modified. It is desirable that the operator $\mathcal{S}_j$ will not use all the points in X_j and F_j but only a subset, i.e. that it has a *finite-support*, i.e. there exists an integer s such that the above equation can be rewritten

$$(x_{k'}^{j+1}, f_{k'}^{j+1}) = \mathcal{S}_j(x_{k-s}^j, \ldots, x_{k+s}^j, f_{k-s}^j, \ldots, f_{k+s}^j).$$

The scheme is termed *stationary* if it does not depend on j, although this definition is weaker than in the linear case, since $\mathcal{S}_j$ is likely to depend on the data.

The notions of convergence and regularity of a sequence of functions obtained through successive refinements derived in Section 2.1 are still perfectly valid in the case of a non-linear subdivision operator.

The idea from [15], further developed in [99], was to study *data-dependent* schemes (which, as we will see in Section 4.2, is relevant for the scheme we propose here). Let us consider a sequence $\{u^j\}_{j\in\mathbb{N}}$, where $u^j = \{u_k^j\}_{k\in\mathbb{Z}} \in \ell_\infty(\mathbb{Z})$, which will act as parameter for the subdivision scheme. At each step, the subdivision scheme depends on u^j, i.e. the data at level $j+1$ denoted by v^{j+1} is computed using the following relation

$$v^{j+1} = \mathcal{S}^j(u^j)v^j.$$

In [15], the case of refinement equations written as

$$v_k^{j+1} = \sum_l a_{k-nl}(u^j)v_l^j, \tag{4.1}$$

was studied, which corresponds to $\mathcal{S}(u^j)$ being a linear operator depending on u^j. Such a scheme that moreover satisfies a recursive dependence relation, i.e. $v^{j+1} = \mathcal{S}(v^j)v^j$ is termed *quasilinear.*

Only the case of *bounded* data-dependent operators has been addressed, i.e. there exists a finite number B such that $\sup_{j\in\mathbb{N}} \left\|\mathcal{S}(u^j)\right\|_\infty \leq B$, where $\|.\|_\infty$ denotes the usual operator norm, which seems a reasonable assumption. Another constraint is that the subdivision rule must be *continuously dependent* on the data, i.e. for every $u, v \in \ell_\infty(\mathbb{Z})$,

$$\|\mathcal{S}(u) - \mathcal{S}(v)\|_\infty \leq C \|u - v\|_\infty ,$$

where C is a positive constant that depends on the quantity $\max\{\|u\|_\infty , \|v\|_\infty\}$. As for eigenanalysis of subdivision schemes (cf. Section 2.3.4), the convergence of a data-dependent scheme relies on the *joint spectral radius*, defined by

$$\rho_\infty(\mathcal{S}) = \overline{\lim_{j\to\infty}} \sup_{(u^0,\ldots,u^{j-1})\in\ell_\infty(\mathbb{Z})^j} \left\|\mathcal{S}(u^{j-1})\cdots\mathcal{S}(u^0)\right\|_\infty^{1/j} .$$

As in Section 2.3.2, the study of the convergence is based on the study of an associated difference scheme. Its existence is shown in the following lemma

Lemma 1. *Let $\mathcal{S}$ be a data-dependent scheme given by equation* (4.1) *and reproducing polynomial up to degree n, then for all $1 \leq p \leq n+1$, there exists an associated scheme $\mathcal{S}_p$ such that*

$$\Delta^p \mathcal{S}(u)v = \mathcal{S}_p(u)\Delta^p v.$$

Proof. Cf. [15] □

In this lemma, the operator Δ is exactly the difference operator seen in Chapter 2. The following theorem provides a link between the joint spectral radius and convergence.

Theorem 17. *Let us consider a subdivision rule $\mathcal{S}$ that follows equation* (4.1)*, and that reproduces constants. If the difference scheme $\mathcal{S}_1$ satisfies $\rho_\infty(\mathcal{S}_1) < 1$, then $\mathcal{S}$ is uniformly convergent and the limit function is $\mathcal{C}^s$ for all $s < -\log_2 \rho_\infty(\mathcal{S}_1)$.*

Proof. Cf. [15] (or [99] although under a slightly different formulation and set of assumptions) □

Of course, computing the spectral radius of such a subdivision operator can be quite challenging. Theorem 17 opens the way to study higher order regularity, through the following sufficient condition.

Theorem 18. *Let us consider a data-dependent rule defined by the refinement equation* (4.1)*, that reproduces polynomials up to degree n. If there exists $0 \leq p \leq n$ such that $\rho_\infty(\mathcal{S}_{p+1}) < 2^{-p}$, then $\mathcal{S}$ is uniformly convergent and the limit function is $\mathcal{C}^s$ for all $s < -\log_2 \rho_\infty(\mathcal{S}_{p+1})$.*

As before, the challenge relies on the existence and computation of the spectral radius of the difference operator.

4.2 Non-linear subdivision using local spherical coordinates

The proposed non-linear subdivision technique described in this section aims at bypassing the adaptations that have to be performed when trying to deal with irregular meshes, while keeping a four-point support in the univariate case. The local coordinate system on which the method is based makes the scheme easily adaptable to triangular meshes, regardless of the valence of the vertices.

4.2.1 Local coordinates system

In order to perform the proposed subdivision technique, we need to define a local coordinates system around each point, as well as several local parameters, which are detailed in this section. Those definitions are illustrated by Figure 4.1. When possible, the level index j will be omitted. We will assume that the samples f_k^j belong to $\mathbb{R}$ although the proposed method can be trivially extended to $\mathbb{R}^d$. Let $\mathcal{R}$ be the canonical basis for $\mathbb{R}^2$. Let us denote by P_k the point of the plane of coordinates (x_k, f_k) in $\mathcal{R}$. In the following, we will denote the Euclidean distance between the points P_k and P_{k+1} by $r_k = \sqrt{(x_{k+1} - x_k)^2 + (f_{k+1} - f_k)^2}$. Since we have to deal with more than one coordinate system, we will add the name of the basis as a superscript when needed to avoid confusion, e.g. $\boldsymbol{v}^{\mathcal{B}}$ will denote the coordinates of vector $\boldsymbol{v}$ in the basis $\mathcal{B}$.

In order to define the angles that will be used in our method, we need to define a local coordinate system at P_k by approximating the tangent and normal vectors at this point. An approximation of the *normal vector* $\boldsymbol{n}_k$ at point P_k can be obtained by averaging the unit normals to the edges $P_{k-1}P_k$ and P_kP_{k+1}, weighted by the edges length r_{k-1} and r_k. The unit normal $\boldsymbol{n}_{k-1,k}$ to the edge $P_{k-1}P_k$ can be expressed (modulo the sign) as

$$\boldsymbol{n}_{k-1,k} = \frac{1}{r_{k-1}} \begin{pmatrix} f_{k-1} - f_k \\ x_k - x_{k-1} \end{pmatrix},$$

which yields

$$\boldsymbol{n}_k = \frac{r_{k-1}\boldsymbol{n}_{k-1,k} + r_k \boldsymbol{n}_{k,k+1}}{\|r_{k-1}\boldsymbol{n}_{k-1,k} + r_k \boldsymbol{n}_{k,k+1}\|}. \tag{4.2}$$

Since

$$r_{k-1}\boldsymbol{n}_{k-1,k} + r_k \boldsymbol{n}_{k,k+1} = \begin{pmatrix} f_{k-1} - f_{k+1} \\ x_{k+1} - x_{k-1} \end{pmatrix},$$

the expression of $\boldsymbol{n}_k$ can be simplified as

$$\boldsymbol{n}_k = \sqrt{1 + {f_k^{[1]_s}}^2} \begin{pmatrix} -f_k^{[1]_s} \\ 1 \end{pmatrix}, \tag{4.3}$$

where $f_k^{[1]_s}$ denotes the *symmetrical* first order finite difference at P_k, i.e.

$$f_k^{[1]_s} = \frac{f_{k+1} - f_{k-1}}{x_{k+1} - x_{k-1}}.$$

The computation of $\boldsymbol{t}_k$ is similar and yields

$$\boldsymbol{t}_k = \frac{1}{\sqrt{1 + {f_k^{[1]_s}}^2}} \begin{pmatrix} 1 \\ f_k^{[1]_s} \end{pmatrix}.$$

Without restricting the problem, we can assume that $f_k^{[1]_s}$ is bounded. Therefore, there exists a unique $\theta_k \in \left(-\frac{\pi}{2}; \frac{\pi}{2}\right)$ such that $f_k^{[1]_s} = \tan\theta_k$ for all k. In fact, θ_k is nothing but the angle between $\boldsymbol{t}_k$ and the horizontal axis in $\mathcal{R}$.

Let us denote the local coordinates system at point P_k by $\mathcal{R}_k = (\boldsymbol{t}_k, \boldsymbol{n}_k)$, for all k and let $\boldsymbol{v}_k$ be the vector from P_k to P_{k+1}. Since we have assumed that the grid is strictly increasing, it is clear that there exists a unique $\gamma_k \in \left(-\frac{\pi}{2}; \frac{\pi}{2}\right)$ such that

$$\boldsymbol{v}_k^{\mathcal{R}} = r_k \begin{pmatrix} \cos\gamma_k \\ \sin\gamma_k \end{pmatrix}.$$

The polar coordinates of P_{k+1} in the canonical basis centered at P_k are (r_k, γ_k). We can now rewrite the expression of $\boldsymbol{v}_k$ in the $\mathcal{R}_k$ basis. There exists a unique $\alpha_k \in (-\pi; \pi)$ such that

$$\boldsymbol{v}_k^{\mathcal{R}_k} = r_k \begin{pmatrix} \cos \alpha_k \\ \sin \alpha_k \end{pmatrix},$$

and α_k, γ_k and θ_k are linked through the following relation

$$\gamma_k = \alpha_k + \theta_k. \tag{4.4}$$

Similarly, we denote by $\boldsymbol{w}_k$ the vector from P_k to P_{k-1} and there exists a unique $\beta_k \in (-\pi; \pi)$ such that

$$\boldsymbol{w}_k^{\mathcal{R}_k} = -r_{k-1} \begin{pmatrix} \cos \beta_k \\ \sin \beta_k \end{pmatrix},$$

and a relation similar to (4.4) holds

$$\gamma_{k-1} = \beta_k + \theta_k. \tag{4.5}$$

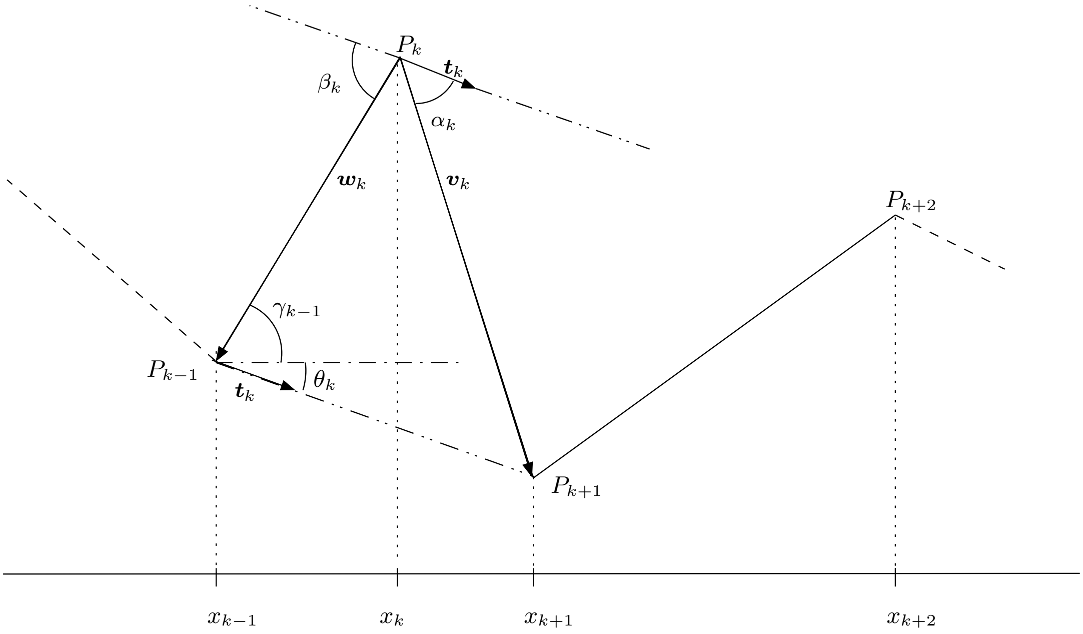


Figure 4.1: Local coordinates system and angles at P_k.

4.2.2 Proposed univariate subdivision scheme

Using the angles and local coordinate system defined above, we can now describe the algorithm to build the level $j + 1$ from the samples at level j. As many popular subdivision algorithms, the proposed method belongs to the class of binary *interpolating* subdivision schemes. Therefore, we will have $x_k^j = x_{2k}^{j+1}$ and $f_k^j = f_{2k}^{j+1}$.

Let us now describe how the new points of level $j + 1$ are obtained. In the following, all steps necessary to compute the midpoint of edge $P_k^j P_{k+1}^j$ will be performed with r_k^j and α_k^j in the $\mathcal{R}_k^j$

coordinate system. A similar development can be made in $\mathcal{R}^j_{k+1}$, with r^j_k and β^j_{k+1}. Let us develop how the point P^{j+1}_{2k+1} (i.e. the "midpoint" of the edge $P^j_k P^j_{k+1}$) is computed.

In $\mathcal{R}^j_k$, the neighboring points of P^j_k are uniquely defined by their local spherical (here polar) coordinates, i.e. by r^j_k and α^j_k. A trivial interpolation, only resulting in a $\mathcal{C}^0$ limit function, would be achieved by taking the middle of the edge $P^j_k P^j_{k+1}$. The coordinates in $\mathcal{R}^j_k$ of such a midpoint would be $\big(\frac{r^j_k}{2}, \alpha^j_k\big)$. Another possible choice would be to use the "middle" of the edge in the parameter domain, i.e. $\big(\frac{r^j_k}{2}, \frac{\alpha^j_k}{2}\big)$. While the choice of $\frac{r^j_k}{2}$ for the first coordinate is rather logical (since we want the newly inserted point's x-coordinate to lie close to the middle of the edge's x-coordinates), the operation to be performed over the α^j_k coordinate is less intuitive. Since the goal is to have a limit function that is the smoothest, i.e. to have the α^j_k and β^j_k become as small as possible as j increases, thus the intuitive choice of $\frac{\alpha^j_k}{2}$ for the second coordinate of the midpoint. While this solution would be elegant, it cannot be applied directly. In fact, the angle α can vary over the whole range $(-\pi;\pi)$ and cases where the principle of having a strictly increasing grid at each level would not be valid may occur. Since the x-coordinate of the new midpoint depends on $\cos\alpha^j_k$, a function having strong variations would lead to values of this quantity close to -1, finally leading to $x^{j+1}_{2k+1} < x^{j+1}_{2k} = x^j_k$, which is obviously an undesirable result. As a consequence, we have chosen to define the "midpoint" as the point of coordinates $\big(\frac{r^j_k}{2}, h(\alpha^j_k)\big)$. The design of function h will be detailed in the next section.

The coordinates $(\hat{x}^{j+1}_{2k+1}, \hat{f}^{j+}_{2k+1})$ of this midpoint in $\mathcal{R}^j_k$ are given by

$$\begin{pmatrix} \hat{x}^{j+1}_{2k+1} - x^j_k \\ \hat{f}^{j+1}_{2k+1} - f^j_k \end{pmatrix}^{\mathcal{R}^j_k} = \frac{r^j_k}{2} \begin{pmatrix} \cos h(\alpha^j_k) \\ \sin h(\alpha^j_k) \end{pmatrix}. \tag{4.6}$$

Another midpoint can be computed in a similar way, using $\mathcal{R}^j_{k+1}$ and β^j_{k+1} instead of $\mathcal{R}^j_k$ and α^j_k, which yields

$$\begin{pmatrix} \tilde{x}^{j+1}_{2k+1} - x^j_{k+1} \\ \tilde{f}^{j+1}_{2k+1} - f^j_{k+1} \end{pmatrix}^{\mathcal{R}^j_{k+1}} = -\frac{r^j_k}{2} \begin{pmatrix} \cos h(\beta^j_{k+1}) \\ \sin h(\beta^j_{k+1}) \end{pmatrix}. \tag{4.7}$$

In order to be able to combine these two ways of computing the midpoint of the edge $P^j_k P^j_{k+1}$, let us rewrite equations (4.6) and (4.7) in $\mathcal{R}$,

$$\begin{aligned} \begin{pmatrix} \hat{x}^{j+1}_{2k+1} - x^j_k \\ \hat{f}^{j+1}_{2k+1} - f^j_k \end{pmatrix}^{\mathcal{R}} &= \frac{r^j_k}{2} \begin{pmatrix} \cos\Big(\theta^j_k + h(\alpha^j_k)\Big) \\ \sin\Big(\theta^j_k + h(\alpha^j_k)\Big) \end{pmatrix}, \\ \begin{pmatrix} \tilde{x}^{j+1}_{2k+1} - x^j_{k+1} \\ \tilde{f}^{j+1}_{2k+1} - f^j_{k+1} \end{pmatrix}^{\mathcal{R}} &= -\frac{r^j_k}{2} \begin{pmatrix} \cos\Big(\theta^j_{k+1} + h(\beta^j_{k+1})\Big) \\ \sin\Big(\theta^j_{k+1} + h(\beta^j_{k+1})\Big) \end{pmatrix}. \end{aligned}$$

The most natural way to take into account these two ways of computing the midpoint is to average the two contributions. Therefore, the final expression of the midpoint in $\mathcal{R}$ will be

$$\begin{aligned} x^{j+1}_{2k+1} = \frac{\hat{x}^{j+1}_{2k+1} + \tilde{x}^{j+1}_{2k+1}}{2} &= \frac{x^j_k + x^j_{k+1}}{2} \\ &\quad + \frac{r^j_k}{4}\Big[\cos\big(\theta^j_k + h(\alpha^j_k)\big) - \cos\big(\theta^j_{k+1} + h(\beta^j_{k+1})\big)\Big], \end{aligned} \tag{4.8}$$

$$\begin{aligned} f^{j+1}_{2k+1} = \frac{\hat{f}^{j+1}_{2k+1} + \tilde{f}^{j+1}_{2k+1}}{2} &= \frac{f^j_k + f^j_{k+1}}{2} \\ &\quad + \frac{r^j_k}{4}\Big[\sin\big(\theta^j_k + h(\alpha^j_k)\big) - \sin\big(\theta^j_{k+1} + h(\beta^j_{k+1})\big)\Big]. \end{aligned} \tag{4.9}$$

Those relations, along with the interpolation rule for even samples fully define the level $j+1$ from the level j.

The proposed scheme is clearly non-linear, due to the sine and cosine terms added to a linear part. A consequence of the method used to compute the x_k^{j+1} is that the grids generated are not uniform. However, numerical results tend to show that the grid gets closer to a uniform grid as the level increases. It is also stationary in the meaning presented in Section 4.1, i.e. the same operator is used to perform subdivision for all samples, although the operator depends on the surrounding data. An obvious property of this scheme is also its ability to reproduce polynomials up to degree 1, since in this case the non-linear part of equations (4.8) and (4.9) will be zero.

As stated in the previous section, the reason for having a function h instead of a pure scalar factor is to avoid pathological cases that may occur when the initial grid has sharp transitions (i.e. values of γ_k^j close to $-\frac{\pi}{2}$ or $\frac{\pi}{2}$). In order to illustrate what occurs in such cases, let us rewrite equation (4.8)

$$x_{2k+1}^{j+1} - x_{2k}^{j+1} = \frac{x_{k+1}^j - x_k^j}{2} + \frac{r_k^j}{4}\Big[\cos\big(\theta_k^j + h(\alpha_k^j)\big) - \cos\big(\theta_{k+1}^j + h(\beta_{k+1}^j)\big)\Big].$$

One can construct cases where the quantity $\cos\big(\theta_k^j + h(\alpha_k^j)\big) - \cos\big(\theta_{k+1}^j + h(\beta_{k+1}^j)\big)$ becomes negative and may lead to a grid that is no longer increasing at the level $j+1$.

The method described above can be generalized to generic plane curves (e.g. using planar polygons as initial data). The computation of the normal vector at each sample point is done using equation (4.3). The closed form derived in equations (4.8) and (4.9) cannot be applied directly since assumptions regarding the monotonicity of the x_k^j have been done. The algorithm 4.1 describes the steps needed to compute the coordinates of the midpoint of each edge. The test performed to determine whether $\|P_k^j M_{k+1}^j\|$ is greater than an ε is useful to avoid pathological cases when computing the angles α_k^j or β_k^j (in our experiments we have chosen $\varepsilon = 10^{-10}$).

Algorithm 4.1: Midpoint computation for plane curves

```
1:  for all edges $P_k^j P_{k+1}^j$ do
2:      Compute the normal vector $\boldsymbol{n}_k^j$ at $P_k^j$ using relation (4.3)
3:      Compute the equation of the line $\mathcal{L}$, normal to $\boldsymbol{n}_k^j$ passing at $P_k^j$
4:      Compute the projection $M_{k+1}^j$ of $P_{k+1}^j$ on $\mathcal{L}$ along $\boldsymbol{n}_k^j$
5:      if $\|P_k^j M_{k+1}^j\| < \varepsilon$ then
6:          $\hat{P}_{2k+1}^{j+1} = \frac{P_k^j + P_{k+1}^j}{2}$
7:      else
8:          Compute the angle $\alpha_k^j = \pm \arctan\left(\frac{\|P_{k+1}^j M_{k+1}^j\|}{\|P_k^j M_{k+1}^j\|}\right)$
9:          Compute the polar coordinates $\left(\frac{r_k^j}{2}, h(\alpha_k^j)\right)$ of $\hat{P}_{2k+1}^{j+1}$ in $\mathcal{R}_k^j$
10:         Compute the coordinates of $\hat{P}_{2k+1}^{j+1}$ in $\mathcal{R}$
11:     end if
12:     Repeat all the steps from line 2 to 11 using $P_{k+1}^j$ as basis to compute $\tilde{P}_{2k+1}^{j+1}$
13:     $P_{2k+1}^{j+1} = \frac{\hat{P}_{2k+1}^{j+1} + \tilde{P}_{2k+1}^{j+1}}{2}$
14: end for
```

4.3 Univariate scheme analysis

In this section, we compute some parameters of the grid at level $j+1$, and use them to design a suitable function h as described in Section 4.2.2 and to analyze the proposed scheme.

4.3.1 Parameters of level $j+1$

Figure 4.2 details the generation of the midpoint M associated to the edge $P_k^j P_{k+1}^j$.

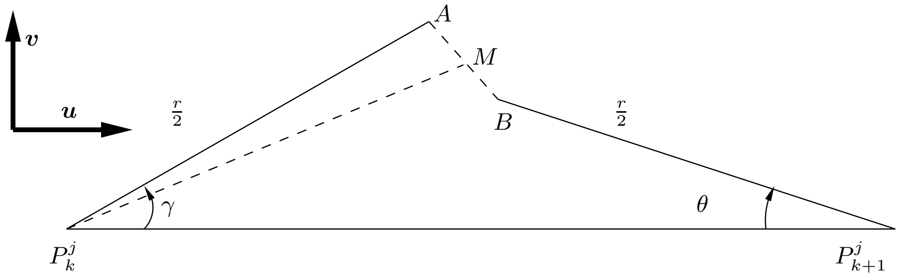


Figure 4.2: Midpoint computation

For each edge, we have to compute the middle of AB. For this purpose, we will consider the orthonormal basis centered in P_k^j, with basis vectors $\boldsymbol{u}, \boldsymbol{v}$. In this coordinate system, if we define r as $r = \|P_k^j P_{k+1}^j\|$, the vectors $\boldsymbol{P_k^j A}$ and $\boldsymbol{P_k^j B}$ can be written as

$$\begin{aligned}\boldsymbol{P_{j,k}A} &= \frac{r}{2}\begin{pmatrix}\cos\gamma\\ \sin\gamma\end{pmatrix}\\ \boldsymbol{P_{j,k}B} &= \frac{r}{2}\begin{pmatrix}2-\cos\theta\\ -\sin\theta\end{pmatrix}.\end{aligned}$$

The minus sign in the y-coordinate of $\boldsymbol{P_k^j B}$ comes from the orientation convention chosen in Figure 4.2. Using these relations, the coordinates of M in this system are

$$\boldsymbol{P_k^j M} = \frac{r}{4}\begin{pmatrix}2-\cos\theta+\cos\gamma\\ \sin\gamma-\sin\theta\end{pmatrix}.$$

If we denote by β the angle between $\boldsymbol{P_k^j M}$ and $\boldsymbol{u}$, the following relation holds,

$$\tan\beta = \frac{\sin\gamma-\sin\theta}{2+\cos\gamma-\cos\theta}. \tag{4.10}$$

The same computations can be done “backward” for the angle between MP_{k+1}^j and $\boldsymbol{u}$ denoted by α,

$$\tan\alpha = \frac{\sin\theta-\sin\gamma}{2+\cos\theta-\cos\gamma}. \tag{4.11}$$

In the canonical basis $\mathcal{R}$, the angles θ and γ can be written as

$$\begin{aligned}\theta &= \theta_k^j + h(\alpha_k^j) - \gamma_k^j\\ \gamma &= \theta_{k+1}^j + h(\beta_{k+1}^j) - \gamma_k^j.\end{aligned}$$

Using equations (4.4) and (4.5), we have

$$\begin{aligned} \theta &= h(\alpha_k^j) - \alpha_k^j \\ \gamma &= h(\beta_{k+1}^j) - \beta_{k+1}^j. \end{aligned} \tag{4.12}$$

Since we have $\gamma_k^j = \theta_{2k+1}^{j+1}$, we also have

$$\gamma_{2k}^{j+1} = \gamma_k^j + \beta = \theta_{2k+1}^{j+1} + \beta_{2k+1}^{j+1} \tag{4.13}$$

$$\gamma_{2k+1}^{j+1} = \gamma_k^j + \alpha = \theta_{2k+1}^{j+1} + \alpha_{2k+1}^{j+1}, \tag{4.14}$$

which yields

$$\begin{aligned} \beta &= \beta_{2k+1}^{j+1} \\ \alpha &= \alpha_{2k+1}^{j+1}. \end{aligned}$$

While on most of the grids $\Gamma = \sup_{j,k} |\gamma_k^j| < \frac{\pi}{2}$, equations (4.13) and (4.14) show clearly that pathological cases may exist where $|\gamma_k^{j+1}|$ could be greater than $\frac{\pi}{2}$, causing the grid at level $j+1$ to be non-strictly increasing anymore. This will be analyzed in the next section.

Finally, using metric relations in triangles, the distances between P_k^j, P_{k+1}^j and M are

$$(r_{2k}^{j+1})^2 = \left(\frac{r_k^j}{4}\right)^2 \Big[6 - 4\cos\theta + 4\cos\gamma - 2\cos(\theta-\gamma)\Big] \tag{4.15}$$

$$(r_{2k+1}^{j+1})^2 = \left(\frac{r_k^j}{4}\right)^2 \Big[6 - 4\cos\gamma + 4\cos\theta - 2\cos(\theta-\gamma)\Big]. \tag{4.16}$$

According to equation (4.12), the bounds of θ and γ are related to the bounds of the function g defined as

$$g : x \longmapsto h(x) - x, \text{ with } x \in (-\pi; \pi). \tag{4.17}$$

4.3.2 Design of h and convergence

As stated in Section 4.2.2, we would like h to be close to a pure scaling of $\frac{1}{2}$. However, in order to avoid the pathological cases described above, g should be 0 when x comes close to π. The initial guess for h that satisfies such requirements is presented in the following equation.

$$h(x) = \begin{cases} x & \text{if } \pi < x \leq -\frac{\pi}{2} \\ -\frac{1}{\pi}\left(x+\frac{\pi}{4}\right)^2 \left[\frac{24}{\pi}\left(x+\frac{\pi}{4}\right)+10\right] + \frac{x}{2} & \text{if } -\frac{\pi}{2} < x < -\frac{\pi}{4} \\ \frac{x}{2} & \text{if } -\frac{\pi}{4} \leq x \leq \frac{\pi}{4} \\ \frac{1}{\pi}\left(x-\frac{\pi}{4}\right)^2 \left[\frac{-24}{\pi}\left(x-\frac{\pi}{4}\right)+10\right] + \frac{x}{2} & \text{if } \frac{\pi}{4} < x < \frac{\pi}{2} \\ x & \text{if } \frac{\pi}{2} \leq x < \pi \end{cases}. \tag{4.18}$$

Function h is a scaling by a factor $\frac{1}{2}$ for small values of x (i.e. $|x| \leq \frac{\pi}{4}$), and the identity when $|x| \geq \frac{\pi}{2}$. Two cubic polynomials provide the $\mathcal{C}^1$ link between these two parts. Figure 4.3 shows the aspect of h.

Using equation (4.18), the extrema of g are easily found. They are located at $x = \pm\frac{5\pi}{18}$, and their value is $\pm\theta_0$ with $\theta_0 = \frac{71\pi}{486}$; thus, $\frac{\pi}{4} < \theta_0 < \frac{\pi}{2}$. Finding the global extrema of α and β is easy. They occur when $\theta = -\gamma = \pm\theta_0$. Therefore,

$$\max_k |\alpha_{2k+1}^{j+1}| = \max_k |\beta_{2k+1}^{j+1}| = \arctan(\sin\theta_0) < \frac{\pi}{4}, \tag{4.19}$$

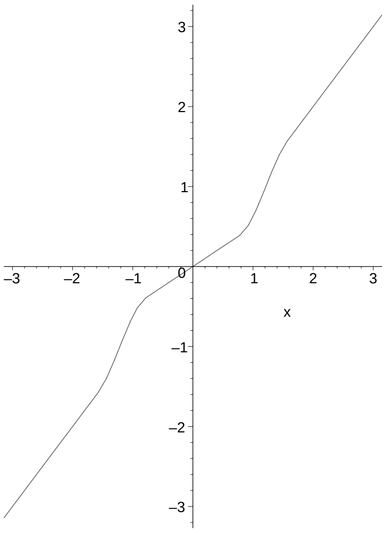


Figure 4.3: Plot of the function h defined by equation (4.18), over the interval $[-\pi;\pi]$

which also means that the following relation holds for all k,

$$\begin{aligned} h(\alpha_{2k+1}^{j+1}) &= \frac{\alpha_{2k+1}^{j+1}}{2} \\ h(\beta_{2k+1}^{j+1}) &= \frac{\beta_{2k+1}^{j+1}}{2}. \end{aligned}$$

In order to study more in details the behavior of γ^{j+1}, especially the potential overflows, i.e. that its absolute value becomes greater than $\frac{\pi}{2}$, we need to analyze further the dependencies of α and β on the local geometry of the grid. In the following development, we will omit the level index when no confusion is possible.

Let us consider a regular grid having a constant step Δx. According to Section 4.2 and equation (4.4), the angle α_k is

$$\alpha_k = \arctan\left(\frac{f_{k+1} - f_k}{x_{k+1} - x_k}\right) - \arctan\left(\frac{f_{k+1} - f_{k-1}}{x_{k+1} - x_{k-1}}\right),$$

i.e.

$$\alpha_k = \arctan\left(\frac{(\Delta f)_k}{(\Delta x)_k}\right) - \arctan\left(\frac{(\Delta f)_k + (\Delta f)_{k-1}}{(\Delta x)_k + (\Delta x)_{k-1}}\right).$$

Since we have imposed that Δx is constant, we can assume $\Delta x = 1$, which leads to

$$\alpha_k = \arctan(\Delta f)_k - \arctan\left(\frac{(\Delta f)_k + (\Delta f)_{k-1}}{2}\right). \tag{4.20}$$

Using equation (4.5), we clearly have

$$\beta_{k+1} = \arctan(\Delta f)_k - \arctan\left(\frac{(\Delta f)_{k+1} + (\Delta f)_k}{2}\right). \tag{4.21}$$

Since $(\Delta f)_{k-1}$, $(\Delta f)_k$ and $(\Delta f)_{k+1}$ are independent, equations (4.20) and (4.21), we have to study the behavior of the function

$$\alpha(x,y) = \arctan x - \arctan\left(\frac{x+y}{2}\right),\ (x,y) \in [-A, A]^2, \tag{4.22}$$

where A is any positive real number. This bounded definition domain implies that the function f has a bounded first derivative. The partial derivatives are

$$\begin{aligned}\frac{\partial\alpha}{\partial x} &= \frac{y^2+2xy-x^2+2}{(1+x^2)\Big((x+y)^2+4\Big)},\\ \frac{\partial\alpha}{\partial y} &= -\frac{2}{(1+x^2)\Big((x+y)^2+4\Big)}.\end{aligned}$$

Since $\frac{\partial\alpha}{\partial y} < 0$ for all $(x,y) \in \mathbb{R}^2$, the extrema of α will be located at $y = \pm A$. We can now study $\frac{\partial\alpha}{\partial x}$, which is zero at $x = y \pm \sqrt{2y^2+2}$. Since the numerator of $\frac{\partial\alpha}{\partial x}$ is a polynomial of degree 2, the extrema are also reached at the zeroes of the polynomial, i.e. we have a minimum at $y - \sqrt{2y^2+2}$ and a maximum at $y + \sqrt{2y^2+2}$. According to the sign of $\frac{\partial\alpha}{\partial y}$, if y is positive, the absolute minimum is reached at $y - \sqrt{2y^2+2}$, and if y is negative, the absolute maximum is reached at $y + \sqrt{2y^2+2}$. Therefore, we can define two functions

$$\begin{aligned}\alpha_M(y) &= \arctan\left(y+\sqrt{2y^2+2}\right) - \arctan\left(y+\frac{\sqrt{2y^2+2}}{2}\right),\ y<0,\\ \alpha_m(y) &= \arctan\left(y-\sqrt{2y^2+2}\right) - \arctan\left(y-\frac{\sqrt{2y^2+2}}{2}\right),\ y>0,\end{aligned}$$

where α_M and α_m denote respectively the absolute maximum and minimum values of α. Let us study the behavior of these two functions with respect to y. We have

$$\frac{\partial\alpha_M}{\partial y} = -\frac{(y^2+1)(4y+3\sqrt{2y^2+2})}{\sqrt{2y^2+2}\left(3+3y^2+2y\sqrt{2y^2+2}\right)^2},$$

and similarly

$$\frac{\partial\alpha_m}{\partial y} = \frac{(y^2+1)(4y-3\sqrt{2y^2+2})}{\sqrt{2y^2+2}\left(3+3y^2-2y\sqrt{2y^2+2}\right)^2}.$$

The sign of these derivatives is tied to the sign of $4y-3\sqrt{2y^2+2}$, which is always negative. Therefore, it is obvious that $\frac{\partial\alpha_M}{\partial y} > 0$ for all $y < 0$ and $\frac{\partial\alpha_m}{\partial y} < 0$ for all $y > 0$. Since $\alpha_M(0) > 0$ and $\alpha_m(0) < 0$, we have $\alpha_M(y) > 0$ for all $y < 0$ and $\alpha_m(y) < 0$ for all $y > 0$.

Using the relations (4.12), (4.13) and (4.14), we can now derive the extrema of the γ_k^{j+1}. Since we are dealing with a regular grid having $\Delta x = 1$, we have

$$\gamma_k = \arctan(\Delta f)_k,$$

therefore leading to $x = \gamma_k$ in equation (4.22). Moreover, $h(x) - x$ is identically zero for $|x| > x_M$, where $x_M = \frac{\pi}{2}$ for the initial proposal of equation (4.18) and $x_M = \frac{5\pi}{8}$ for the function defined in equation (4.33). The conditions for $|\alpha(x,y)| > \frac{\pi}{2}$ can be summarized as

$$\begin{aligned} y &> -x - \frac{2}{x},\ x<0,\ \text{region } \boxed{1} \text{ in Figure 4.4}\\ y &< -x - \frac{2}{x},\ x>0,\ \text{region } \boxed{5} \text{ in Figure 4.4.}\end{aligned} \tag{4.23}$$

A similar study can be performed to find the domain on which we have $|\alpha(x,y)| < \frac{\pi}{4}$, i.e. for the original h function defined in equation (4.18)

$$\begin{aligned} y &> -\frac{x^2-x+2}{x+1}, \ x > -1, \\ y &< \frac{x^2+x+2}{1-x}, \ x < 1. \end{aligned} \tag{4.24}$$

In addition to these conditions, it is clear that $\alpha(x,y) > 0$ when $y < x$. The aspect of the domain defined by the conditions of equation (4.23) and equation (4.24) is shown in Figure 4.4.

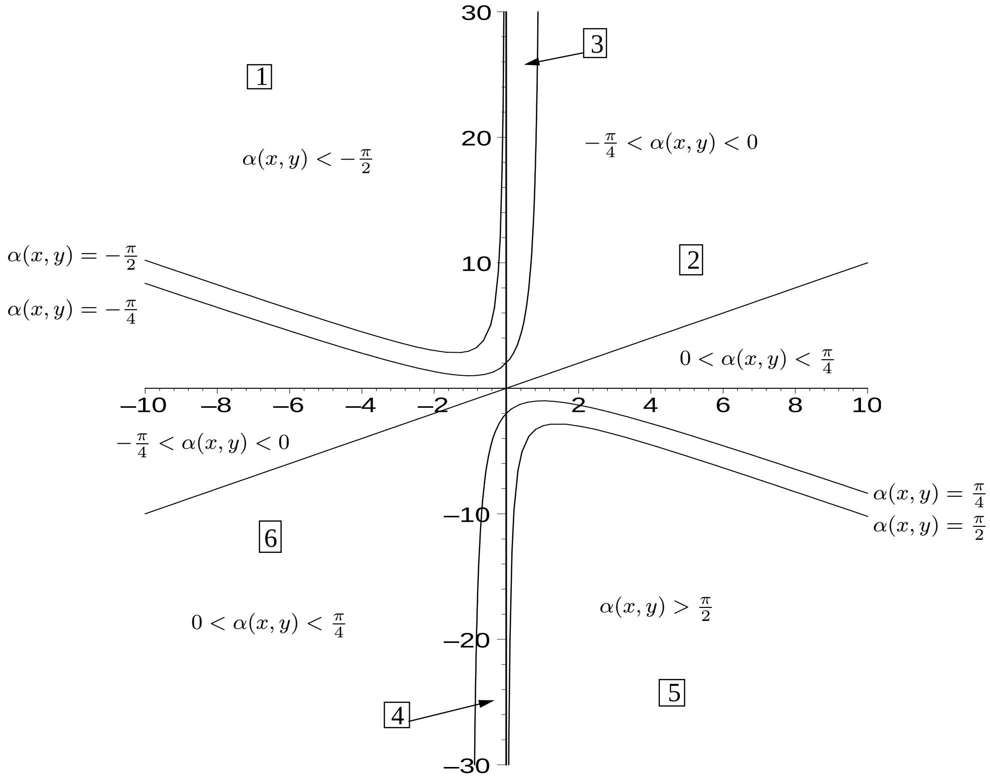


Figure 4.4: Illustrations of conditions given by equation (4.23) and equation (4.24), shown for $x \in [-10; 10]$.

As the quantities defined in equation (4.10) and equation (4.11), which are in fact α_{2k+1}^{j+1} and β_{2k+1}^{j+1} depend on two parameters denoted by θ and γ which are given by equation (4.12). As implied by equation (4.20) and equation (4.21), we need to study jointly two bivariate functions,

$$\begin{aligned} \alpha(x,y) &= \arctan x - \arctan\frac{x+y}{2}, \\ \beta(x,z) &= \arctan x - \arctan\frac{x+z}{2}, \end{aligned}$$

having one common parameter x (which corresponds to $(\Delta f)_k$) and two independent variables, denoted by y (which corresponds to $(\Delta f)_{k-1}$) and z (which corresponds to $(\Delta f)_{k+1}$). According to Figure 4.4, if x is constant, there are several cases to study at each value, depending on the position of the y and z variables.

As seen before, the absolute extrema of equation (4.10) and equation (4.11), which are reached when $\theta = -\gamma = \pm\theta_0$, i.e. when $g(\alpha(x,y)) = \pm\theta_0$ and $g(\beta(x,z)) = \mp\theta_0$. Therefore, the function α

and β have to satisfy simultaneously

$$\begin{aligned}\alpha(x,y) &= \pm\alpha_0,\\ \beta(x,z) &= \mp\alpha_0.\end{aligned}$$

Let us solve the case $\alpha(x,y) = \alpha_0$ and $\beta(x,z) = -\alpha_0$, since the other is trivially inferred from this one. The solutions are

$$\begin{aligned}y &= \frac{-x^2\tan\alpha_0 + x - 2\tan\alpha_0}{x\tan\alpha_0 + 1},\ x > -\frac{1}{\tan\alpha_0},\\ z &= \frac{x^2\tan\alpha_0 + x + 2\tan\alpha_0}{1 - x\tan\alpha_0},\ x < \frac{1}{\tan\alpha_0}.\end{aligned}$$

Clearly, the two conditions can be met simultaneously only when $|x| < \frac{1}{\tan\alpha_0}$, which in our case yields $|x| < \tan\frac{5\pi}{18} \approx 1.192$, which is not a problem for the bounds of γ^{j+1}.

Let us now consider the case $x < -1$, which implies that we will focus on the absolute lower bounds in equation (4.10) and equation (4.11). The different possibilities are studied below, the boxed numbers (e.g. $\boxed{1}$) referring to a specific region of the plane in Figure 4.4.

1. If y and z are in region $\boxed{1}$, we have $\theta = \gamma = 0$, which leads to $\gamma_{k'}^{j+1} = \gamma_k^j$.

2. If either y or z is in region $\boxed{1}$ but the other is in regions $\boxed{2}$ or $\boxed{6}$ (we will assume that y is in $\boxed{2}$ or $\boxed{6}$, but the reasoning is identical for the other case), we have

   $$-\frac{\pi}{4} \le \alpha(x,y) \le \frac{\pi}{2} + \arctan x < \frac{\pi}{4},\ \text{since } x < -1.$$

   Given the assumption z is in region $\boxed{1}$, γ is equal to 0. According to the bound of $\alpha(x,y)$ over this domain, $g(\alpha) = -\frac{\alpha}{2}$, which leads to

   $$-\frac{1}{2}\arctan x - \frac{\pi}{4} \le g(\alpha(x,y)) = \theta \le \frac{\pi}{8}.$$

   Therefore, the lower bound for α_{2k+1}^{j+1} and β_{2k+1}^{j+1} (for values of x large enough) is equal to

   $$\arctan\frac{-\sin\frac{\pi}{8}}{3-\cos\frac{\pi}{8}} < 0.$$

   In this case, if we want γ_k^{j+1} to remain bounded, additional precautions have to be taken.

3. If either y or z is in region $\boxed{1}$ but the other is in region $\boxed{3}$, the conclusions are identical to the one derived in the previous item, except that the lower bound is equal to

   $$\arctan\frac{-\sin\theta_0}{1+\cos\theta_0},$$

   which is also negative (and worse than the previous one)

4. If both y and z are in region $\boxed{3}$, a negative bound is also possible (the same as the above case is valid if we consider the boundary of the domain).

5. If either y or z is in $\boxed{3}$ and the other is in $\boxed{2}$ or $\boxed{6}$, the lower bound is given by (for values of x large enough)

   $$\frac{-\sin\theta_0 + \sin\left(\frac{\pi}{4} + \frac{1}{2}\arctan x\right)}{2+\cos\theta_0 - \cos\left(\frac{\pi}{4} + \frac{1}{2}\arctan x\right)},$$

   which is also negative.

6. If y and z both are in regions $\boxed{2}$ or $\boxed{6}$, the lower bound is given by

$$\frac{-\sin\frac{\pi}{8}+\sin\left(\frac{\pi}{4}+\frac{1}{2}\arctan x\right)}{2+\cos\frac{\pi}{8}-\cos\left(\frac{\pi}{4}+\frac{1}{2}\arctan x\right)},$$

   which is also negative.

An identical analysis can be performed for $x > 1$, with identical conclusions with respect to the precautions to ensure that $\gamma_k^{j+1} < \frac{\pi}{2}$. To avoid such limit cases, the upper bound Γ is set to $\frac{\pi}{2}-\varepsilon$, with $\varepsilon > 0$. This implies that when the angles of level $j+1$ are computed using equations (4.13) and (4.14) are out of the bounds, the following rule is applied if $\gamma_{2k}^{j+1} > \Gamma$ (the case $\gamma_{2k}^{j+1} < -\Gamma$ is easily derived from this)

$$\begin{aligned}\gamma_{2k}^{j+1} &= \Gamma\\ r_{2k}^{j+1} &= r_{2k+1}^{j+1} = \frac{r_k^j}{2\cos(\Gamma-\gamma_k^j)}.\end{aligned}$$

The cases $|\gamma_{2k+1}^{j+1}| > \Gamma$ are also handled in a similar way

$$\begin{aligned}\gamma_{2k+1}^{j+1} &= \Gamma\\ r_{2k+1}^{j+1} &= r_{2k}^{j+1} = \frac{r_k^j}{2\cos(\Gamma-\gamma_k^j)}.\end{aligned}$$

In the following, we will limit ourselves to the "regular" cases, i.e. when the hard limitation of γ_k^{j+1} does not need to be used. According to the absolute bounds of α_{2k+1}^{j+1} and β_{2k+1}^{j+1} derived in equation (4.19), $|\Gamma-\gamma_k^j| \leq \max_k \alpha_k^{j+1}$, which yields the following bound on $r_M^{j+1} = \max_k r_k^{j+1}$ when this limitation is used :

$$r_M^{j+1} < r_M^j \frac{\sqrt{1+\sin^2\theta_0}}{2}. \tag{4.25}$$

The other angles, namely α_{2k}^{j+1} and β_{2k}^{j+1} unfortunately do not follow the same rule. Equations (4.4) and (4.5) yield

$$\alpha_{2k}^{j+1} - \beta_{2k}^{j+1} = \gamma_{2k}^{j+1} - \gamma_{2k-1}^{j+1}, \tag{4.26}$$

and a law in the triangle $P_{2k-1}^{j+1}P_k^jP_{2k+1}^{j+1}$ also gives the following relation,

$$\tan\left(\frac{\alpha_{2k}^{j+1}+\beta_{2k}^{j+1}}{2}\right) = \frac{r_{2k}^{j+1}-r_{2k-1}^{j+1}}{r_{2k}^{j+1}+r_{2k-1}^{j+1}}\tan\left(\frac{\gamma_{2k}^{j+1}-\gamma_{2k-1}^{j+1}}{2}\right). \tag{4.27}$$

The bounds on r_{2k}^{j+1} and $r_{2k\pm1}^{j+1}$ are identical and can be inferred from equations (4.15) and (4.16). Let us focus on the bounds of r_{2k}^{j+1}, since the same reasoning can be performed for r_{2k+1}^{j+1}. Let us denote by σ the ratio $8\left(\frac{r_{2k}^{j+1}}{r_k^j}\right)^2$. Obviously, equation (4.15) yields

$$\sigma(\theta,\gamma) = 3 - 2\cos\theta + 2\cos\gamma - \cos(\theta-\gamma).$$

The partial derivatives of σ are

$$\begin{aligned}\frac{\partial\sigma}{\partial\theta} &= 2\sin\theta - \sin(\gamma-\theta)\\ \frac{\partial\sigma}{\partial\gamma} &= 2\sin\gamma - \sin(\theta-\gamma).\end{aligned}$$

The second derivatives of σ have a constant sign over the interval $I_{\theta_0} = [-\theta_0; \theta_0]$,

$$\begin{aligned}
\frac{\partial^2\sigma}{\partial\theta^2} &= 2\cos\theta + \cos(\gamma - \theta) > 0 \\
\frac{\partial^2\sigma}{\partial\theta\partial\gamma} &= -\cos(\gamma - \theta) < 0 \\
\frac{\partial^2\sigma}{\partial\gamma^2} &= -2\cos\gamma + \cos(\gamma - \theta) < 0.
\end{aligned}$$

The inequality concerning $\frac{\partial^2\sigma}{\partial\gamma^2}$ holds because over I_{θ_0}, $-2\cos\gamma \le -2\cos\theta_0$, which leads to $\frac{\partial^2\sigma}{\partial\gamma^2} \le -2\cos\theta_0 + 1 < 0$.

Let us try to solve $\frac{\partial\sigma}{\partial\theta} = 0$ and $\frac{\partial\sigma}{\partial\gamma} = 0$

$$\begin{aligned}
\frac{\partial\sigma}{\partial\theta} = 0 &\implies \tilde{\theta} = \arctan\left(\frac{\sin\gamma}{2 + \cos\gamma}\right) \\
\frac{\partial\sigma}{\partial\gamma} = 0 &\implies \tilde{\gamma} = \arctan\left(\frac{\sin\theta}{2 + \cos\theta}\right)
\end{aligned}$$

We can see that $\gamma = \theta = 0$ satisfy the two above conditions. However, this is not a global minimum, since the determinant of the Hessian matrix is equal to

$$\det\mathcal{S} = \det\begin{pmatrix} \frac{\partial^2\sigma}{\partial\theta^2} & \frac{\partial^2\sigma}{\partial\theta\partial\gamma} \\ \frac{\partial^2\sigma}{\partial\theta\partial\gamma} & \frac{\partial^2\sigma}{\partial\gamma^2} \end{pmatrix} = -4\cos\theta\cos\gamma + 2\cos(\theta - \gamma)(\cos\theta - \cos\gamma).$$

Clearly, over I_{θ_0}, we have

$$\det\mathcal{S} \le -4\cos^2\theta_0 + 2(1 - \cos\theta_0) < 0,$$

which implies that $\theta = \gamma = 0$ is not a extremum. As a consequence, it is likely that the min./max. values are reached on the boundaries of $I_{\theta_0} \times I_{\theta_0}$. The upper bound r_M^{j+1} is reached when $\theta = -\gamma = \pm\theta_0$ is

$$r_M^{j+1} \le \frac{r_M^j}{4}\sqrt{6 - 2\cos(2\theta_0)} = r_M^j \frac{\sqrt{1 + \sin^2\theta_0}}{2} \approx 0.5469\, r_M^j. \tag{4.28}$$

which is identical to the bound derived in equation (4.25). The quantity $r_m^{j+1} = \min_k r_k^{j+1}$ can also be computed. It is reached for $\gamma = \pm\theta_0$. In this case

$$\sigma = 3 - 2\cos\theta + 2\cos\theta_0 - \cos(\theta - \theta_0),$$

and

$$\frac{\partial\sigma}{\partial\theta} = 2\sin\theta - \sin(\theta - \theta_0),$$

which is 0 at $\tilde{\theta}(\theta_0) = \arctan\left(\frac{\sin\theta_0}{2+\cos\theta_0}\right)$. Since $\frac{\partial\sigma}{\partial\theta}$ is increasing over I_{θ_0}, $\sigma\big(\tilde{\theta}(\theta_0), \theta_0\big)$ is the minimum value, which can be also obtained at $\sigma\big(-\tilde{\theta}(\theta_0), -\theta_0\big)$. Finally,

$$\min_{(\theta,\gamma)\in I_{\theta_0}^2} \sigma = 3 + 2\cos\theta_0 - \sqrt{5 + 4\cos\theta_0}, \tag{4.29}$$

which approximately gives,

$$r_m^{j+1} \approx 0.4825 r_m^j.$$

Theorem 19. *The proposed scheme converges uniformly toward a limit function.*

Proof. Using the above parameters, we can now prove that the scheme converges uniformly toward a limit function, using Theorem 1. Let us focus on the quantity $\mathcal{B}_j = \|f^{j+1} - f^j\|_\infty$, in the case of the spherical subdivision. Since we are in the framework of interpolating subdivision, and using the notations of the previous section we have

$$\begin{aligned}\mathcal{B}_j &= \sup_k \left| f_{2k+1}^{j+1} - f_k^j - f_k^{j\,[1]}(x_{2k+1}^{j+1} - x_k^j) \right| \\ &= \sup_k \left| \frac{r_k^j}{4} \sin\left(\frac{\gamma+\theta}{2}\right) - \tan\gamma_k^j \frac{r_k^j}{4} \cos\left(\frac{\gamma+\theta}{2}\right) \right| .\end{aligned}$$

This expression can be bounded by

$$\mathcal{B}_j \leq \frac{r_k^j}{4} \cos\left(\frac{\gamma+\theta}{2}\right) \left| \tan\left(\frac{\gamma+\theta}{2}\right) - \tan\gamma_k^j \right| ,$$

which leads to :

$$\mathcal{B}_j \leq \frac{r_M^j}{4} \left(\tan\theta_0 + \tan\Gamma\right) . \tag{4.30}$$

Using (4.28), it is obvious that $r_M^j \leq r_M^0 \delta^j$, with $\delta = \frac{\sqrt{1+\sin^2\theta_0}}{2} < 1$, which leads to

$$\mathcal{B}_j \leq \delta^j \frac{r_M^0}{4} \left(\tan\theta_0 + \tan\Gamma\right) , \tag{4.31}$$

which concludes the proof, since $\delta < 1$ and the other term is finite. □

A similar analysis can be derived if a different function h is chosen. Keeping the principle of having a pure scaling by a factor $\frac{1}{2}$ around the origin and the identity when being close to the bounds, an alternate way of connecting these two components can be chosen, as follows

$$h(x) = \begin{cases} x & \text{if } \pi < x \leq -\frac{\pi}{2} - \varepsilon \\ \frac{1}{8\varepsilon}\left(x^2 + (10\varepsilon + \pi)x + \frac{4\varepsilon^2+\pi^2+4\pi\varepsilon}{4}\right) & \text{if } -\frac{\pi}{2} - \varepsilon < x < -\frac{\pi}{2} + \varepsilon \\ \frac{3}{2}x + \frac{\pi}{4} & \text{if } -\frac{\pi}{2} + \varepsilon \leq x \leq -\frac{\pi}{4} - \varepsilon \\ \frac{1}{4\varepsilon}\left(-x^2 + (4\varepsilon - \frac{\pi}{2})x - \frac{\pi^2-8\pi\varepsilon+16\varepsilon^2}{16}\right) & \text{if } -\frac{\pi}{4} - \varepsilon < x < -\frac{\pi}{4} + \varepsilon \\ \frac{x}{2} & \text{if } -\frac{\pi}{4} + \varepsilon \leq x \leq \frac{\pi}{4} - \varepsilon \\ \frac{1}{4\varepsilon}\left(x^2 + (4\varepsilon - \frac{\pi}{2})x + \frac{\pi^2-8\pi\varepsilon+16\varepsilon^2}{16}\right) & \text{if } \frac{\pi}{4} - \varepsilon < x < \frac{\pi}{4} + \varepsilon \\ \frac{3}{2}x - \frac{\pi}{4} & \text{if } \frac{\pi}{4} + \varepsilon \leq x \leq \frac{\pi}{2} - \varepsilon \\ \frac{1}{8\varepsilon}\left(-x^2 + (10\varepsilon + \pi)x - \frac{4\varepsilon^2+\pi^2+4\pi\varepsilon}{4}\right) & \text{if } \frac{\pi}{2} - \varepsilon < x < \frac{\pi}{2} + \varepsilon \\ x & \text{if } \frac{\pi}{2} + \varepsilon \leq x < \pi \end{cases} . \tag{4.32}$$

This function is built on the same piecewise parts that are in the initial proposition of equation (4.18), i.e. $h(x) = \frac{x}{2}$ if $|x| \leq \frac{\pi}{4}$ and $h(x) = x$ if $|x| \geq \frac{\pi}{2}$. The polynomial junction between the linear part is replaced by a linear part. In order for h to be a $\mathcal{C}^1$ function, the junctions of the linear parts around x_0 are achieved using a polynomial between $x_0 - \varepsilon$ and $x_0 + \varepsilon$. Since $x_0 \in \left\{\pm\frac{\pi}{4}, \pm\frac{\pi}{2}\right\}$, ε has to satisfy $0 < \varepsilon \leq \frac{\pi}{8}$ ($\varepsilon = 0$ is possible, but the resulting h is not $\mathcal{C}^1$). The associated function g has extrema at $x = \pm\frac{\pi}{4}$, which implies that the maximum and minimum values of g are $\pm\theta_0$, with $\theta_0 = \frac{\pi}{8} - \frac{\varepsilon}{4}$. Since a smaller value of θ_0 leads to tighter bounds in equations (4.28), and (4.29), the optimal value of ε is $\varepsilon = \frac{\pi}{8}$, which leads to a simpler expression of h, as shown in equation (4.33), since the linear parts between $\pm\frac{\pi}{4} \pm \varepsilon$ and $\pm\frac{\pi}{2} \mp \varepsilon$ are not needed anymore.

$$h(x) = \begin{cases} x & \text{if } \pi < x \leq -\frac{5\pi}{8} \\ \frac{1}{64\pi}\left(64x^2 + 144\pi x + 25\pi^2\right) & \text{if } -\frac{5\pi}{8} < x < -\frac{3\pi}{8} \\ -\frac{1}{32\pi}\left(64x^2 + \pi^2\right) & \text{if } -\frac{3\pi}{8} \leq x < -\frac{\pi}{8} \\ \frac{x}{2} & \text{if } -\frac{\pi}{8} \leq x \leq \frac{\pi}{8} \\ \frac{1}{32\pi}\left(64x^2 + \pi^2\right) & \text{if } \frac{\pi}{8} < x \leq \frac{3\pi}{8} \\ \frac{1}{64\pi}\left(-64x^2 + 144\pi x - 25\pi^2\right) & \text{if } \frac{3\pi}{8} < x < \frac{5\pi}{8} \\ x & \text{if } \frac{5\pi}{8} \leq x < \pi \end{cases} . \tag{4.33}$$

The function defined by equation (4.33) is shown in Figure 4.5

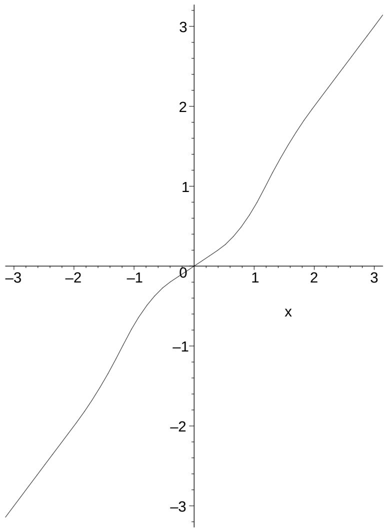


Figure 4.5: Alternative for h defined in equation (4.33) over $[-\pi, \pi]$

The analysis preformed with $\alpha(x, y)$ previously is valid. The analogous of conditions from equation (4.23) and equation (4.24) are

$$\begin{aligned} y &< \frac{-x^2 \tan\frac{5\pi}{8} + x - 2\tan\frac{5\pi}{8}}{x\tan\frac{5\pi}{8} + 1}, \; x > -\frac{1}{\tan\frac{5\pi}{8}}, \\ y &> \frac{x^2 \tan\frac{5\pi}{8} + x + 2\tan\frac{5\pi}{8}}{1 - x\tan\frac{5\pi}{8}}, \; x < \frac{1}{\tan\frac{5\pi}{8}}, \end{aligned} \tag{4.34}$$

to have $|\alpha(x, y)| > \frac{5\pi}{8}$, and

$$\begin{aligned} y &> \frac{-x^2 \tan\frac{\pi}{8} + x - 2\tan\frac{\pi}{8}}{x\tan\frac{\pi}{8} + 1}, \; x > -\frac{1}{\tan\frac{\pi}{8}}, \\ y &< \frac{x^2 \tan\frac{\pi}{8} + x + 2\tan\frac{\pi}{8}}{1 - x\tan\frac{\pi}{8}}, \; x < \frac{1}{\tan\frac{\pi}{8}}, \end{aligned} \tag{4.35}$$

to have $|\alpha(x, y)| < \frac{\pi}{8}$. The aspect of the domains are shown in Figure 4.6.
The study regarding the bounds of γ_k^{j+1} in this case is adapted straightforwardly from the one carried for the other choice of h.

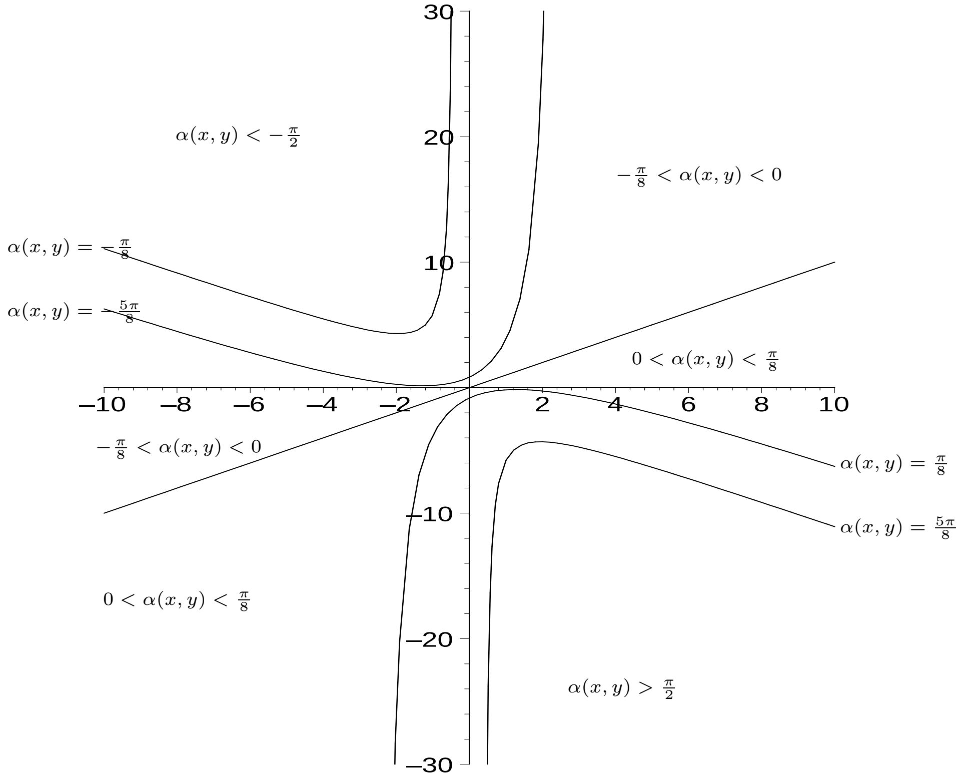


Figure 4.6: Illustrations of conditions given by equation (4.35) and equation (4.34), shown for $x \in [-10; 10]$.

Finally, the absolute extrema of g are $\pm\theta_0$ with $\theta_0 = \frac{3\pi}{32}$. The bound on r_M^{j+1} of equation (4.28) becomes

$$r_M^{j+1} \lessapprox 0.5206\, r_M^j. \tag{4.36}$$

Finally the bound on r_m^{j+1} becomes

$$0.4928\, r_m^j \lessapprox r_m^{j+1}.$$

The convergence study remains valid for the function h proposed in equation (4.33). Since the bound of equation (4.36) is tighter than the one of equation (4.28), the scheme is likely to converge faster using the alternative function for h.

Regarding higher-order analysis, the only straightforward property is that the first derivative remains bounded (by $\tan\Gamma$) throughout the levels. However, deriving finer estimates, and proving convergence of the first derivative is a much more complex task we will not address here.

4.4 Extension to surfaces

As for the univariate case presented in Section 4.2.2, it is possible to define a local coordinate system for a surface. We will use the definitions of Section 3.3, restricted to manifold triangulated discrete surfaces defined by a net $\mathcal{N} = (V, E, F)$. We will also restrict ourselves to *orientable* surfaces, i.e. all the faces surrounding a vertex have normal vectors pointing to the "same side" of the surfaces (a typical example of non-orientable surface is the Möbius strip or the Klein bottle). The local coordinates system at each vertex can be easily defined provided that the normal vector to the surface can be estimated at each vertex. Since we are dealing with an orientable surface, all the faces surrounding vertex v_i have a normal vector oriented in a consistent way. The normal vector $\boldsymbol{n}_{v_i}$ at vertex v_i can be estimated by averaging the normal vectors of the faces having v_i as vertex, weighted by the area of each triangle. We will denote by $E_{v_i} \subset E$ the set of edges having v_i as vertex (i.e. the 1-ring of v_i). If we denote by $F_{v_i} \subset F$ the set of faces having v_i as vertex, by $\boldsymbol{n}_f$ the unit normal to the face f and by $|f|$ the area of face f, the expression of $\boldsymbol{n}_{v_i}$ is

$$\boldsymbol{n}_{v_i} = \frac{\sum_{f \in F_{v_i}} |f| \boldsymbol{n}_f}{\| \sum_{f \in F_{v_i}} |f| \boldsymbol{n}_f \|},$$

which is analogous to the expression (4.2) used in the univariate case. At each vertex v_i, it is now easy to define a local coordinate system $\mathcal{R}_{v_i} = (\boldsymbol{t}_{v_i}, \boldsymbol{u}_{v_i}, \boldsymbol{n}_{v_i})$, where $(\boldsymbol{t}_{v_i}, \boldsymbol{u}_{v_i})$ is an orthonormal basis of the tangent plane at vertex v_i. Let v_k be a vertex sharing an edge with v_i. For each edge $\{v_i, v_k\}$ we define $\boldsymbol{v}_{i,k}$ as the vector from v_i to v_k, as well as the corresponding unit vector $\boldsymbol{w}_{i,k} = \frac{1}{r_{i,k}} \boldsymbol{v}_{i,k}$, where $r_{i,k} = \|\boldsymbol{v}_{i,k}\|$. Since $\boldsymbol{w}_{i,k}$ is a unit vector, we can define two angles $\theta_{i,k}$ and $\phi_{i,k}$ which are the usual spherical coordinates

$$\boldsymbol{w}_{i,k}^{\mathcal{R}_{v_i}} = \begin{pmatrix} \cos\phi_{i,k} \sin\theta_{i,k} \\ \sin\phi_{i,k} \sin\theta_{i,k} \\ \cos\theta_{i,k} \end{pmatrix}.$$

Therefore, the spherical coordinates of $\boldsymbol{v}_{i,k}$ in $\mathcal{R}_{v_i}$ are $(r_{i,k}, \theta_{i,k}, \phi_{i,k})$. Figure 4.7 illustrates the various notations proposed.

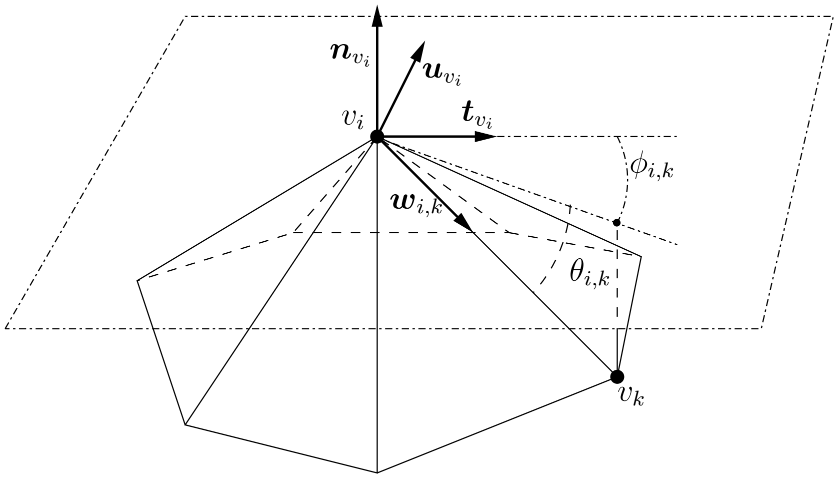


Figure 4.7: Local coordinate system around vertex v_i. (NB: the "dash-dotted" lines all belong to the same plane)

Using these definitions, we can describe the subdivision method for triangulated surfaces. Let us denote by $\mathcal{N}^j = (V^j, E^j, F^j)$ the discrete surface at level j. The midpoint computation is closely related to the method described in Section 4.2.2. Let us consider a vertex v_i^j belonging to V^j. For each vertex $v_k^j \in E_{v_i^j}$, the spherical coordinates of the vector $\boldsymbol{v}_{i,k}^j$ in $\mathcal{R}_{v_i^j}$ are $(r_{i,k}^j, \theta_{i,k}^j, \phi_{i,k}^j)$. Let us define the vector $\hat{\boldsymbol{v}}_{i,k}^{j+1}$ such that its spherical coordinates in $\mathcal{R}_{v_i^j}$ are

$$\hat{\boldsymbol{v}}_{i,k}^{j+1} = \left(\frac{r_{i,k}^j}{2}, h(\theta_{i,k}^j), \phi_{i,k}^j \right). \tag{4.37}$$

According to the definition of $\boldsymbol{v}^j_{i,k}$, there exists a unique vertex $\hat{v}^{j+1}_{i,k}$ such that $\hat{\boldsymbol{v}}^{j+1}_{i,k}$ is the vector from v^j_i to $\hat{v}^{j+1}_{i,k}$. A similar computation can be done by writing the spherical coordinates of the vector $\boldsymbol{v}^j_{k,i}$ in $\mathcal{R}_{v^j_k}$, which leads to the definition of $\tilde{\boldsymbol{v}}^{j+1}_{k,i}$

$$\tilde{\boldsymbol{v}}^{j+1}_{k,i} = \left(\frac{r^j_{i,k}}{2}, h(\theta^j_{k,i}), \phi^j_{k,i} \right). \tag{4.38}$$

This relation leads to the definition of a unique $\tilde{v}^{j+1}_{k,i}$ such that $\tilde{\boldsymbol{v}}^{j+1}_{k,i}$ is the vector from v^j_k to $\tilde{v}^{j+1}_{k,i}$. An interesting property of the proposed method, derived from equations (4.37) and (4.38) is that the scheme is locally invariant with respect to a rotation around the normal vector, since the ϕ angles are not modified in the subdivision process.

Finally, the coordinates of the new vertex $v^{j+1}_{i,k}$ in the canonical basis of $\mathbb{R}^3$, inserted between v^j_i and v^j_k is computed using the following relation

$$v^{j+1}_{i,k} = \frac{\hat{v}^{j+1}_{i,k} + \tilde{v}^{j+1}_{k,i}}{2}.$$

The actual computation of the midpoint can be achieved using an algorithm close to algorithm 4.1, except the projection has to be done into the tangent plane instead of the tangent line.

When boundary edges are encountered, the proposed method can still be applied although the definition of normal vectors has to be modified. Still the normal vector at a vertex belonging to a boundary edge is computed using relation (4.2). The difference lies in the definition of the normal vertex for a boundary edge. Let us consider a triangle formed by vertices (v_k, v_l, v_m), where $v_k v_l$ is a boundary edge. If we denote by $\boldsymbol{n}$ the normal vector to the considered triangle, we define the normal $\boldsymbol{n}_{k,l}$ to edge $v_k v_l$ by

$$\boldsymbol{n}_{k,l} = \frac{\boldsymbol{n}_{k,l} \times \boldsymbol{n}}{\|\boldsymbol{n}_{k,l} \times \boldsymbol{n}\|},$$

which is a unit vector orthogonal to both $\boldsymbol{n}$ and $\boldsymbol{v}_{k,l}$. In case when $\boldsymbol{n}$ and $\boldsymbol{n}_{k,l}$ are colinear, an alternate way of computing the normals has to be chosen, for instance by averaging the normals of the neighboring faces, using an adaptation of equation (4.2).

4.5 Conclusion

In this chapter, we have introduced a new class of subdivision method, suitable for univariate data, plane curves and discrete triangulated surfaces. Unlike linear techniques, adaptation to surfaces from the univariate case is straightforward. The proposed method converges toward a continuous limit function, and has bounded first order derivative. However, given the complexity of the analysis (linked to the high number of local parameters of the grid/surface), no proof is available for higher-order convergence. As a consequence, the next chapter will study the proposed scheme using numerical criteria and compare it to known linear schemes, having an identical support.

5

Numerical study of subdivision schemes

In Chapter 4, we have introduced a non-linear subdivision scheme. Although conceptually simple, the scheme leads to rather complicated computations when trying to check its convergence and the regularity of the limit function. The few theoretical tools developed for non-linear subdivision schemes are not directly applicable. Hence, the need to develop tools to study numerically the proposed subdivision scheme, in order to derive conjectures with respect to the behavior of the subdivision technique. This chapter is organized as follows: numerical criteria to study univariate schemes will be presented in Section 5.1, and partially extended to discrete surfaces in Section 5.2. An extensive numerical comparison between the proposed non-linear scheme and linear methods having an identical support will be presented in Section 5.3 for the univariate case and in Section 5.4 for surface subdivision.

5.1 Numerical criteria for univariate schemes

In this section we introduce numerical tools to check the critical behavior of a subdivision, i.e. convergence (in Section 5.1.1), regularity of the limit function (in Section 5.1.2), and approximation order (in Section 5.1.3).

5.1.1 Convergence

Again, Theorem 1 can be used a basis to derive a convergence criterion. According to this theorem, a sufficient condition for the sequence of piecewise linear functions $\{f^j\}_{j\in\mathbb{N}}$ to converge uniformly toward a limit function is to have

$$\left\|f^j - f^{j-1}\right\|_\infty < \beta\alpha^{j-1}, \tag{5.1}$$

with $\beta > 0$ and $0 < \alpha < 1$. Taking the logarithm of this conditions yields

$$\log\left\|f^j - f^{j-1}\right\|_\infty < \log\beta + (j-1)\log\alpha.$$

According to this relation, the behavior of $\log \left\| f^j - f^{j-1} \right\|_\infty$ should be linear with respect to j and by computing this quantity for each level, we should be able to derive an upper bound for α and β. In the case of interpolating subdivision, the maximum of $|f^j - f^{j-1}|$ is reached at odd breakpoints, i.e.

$$\left\| f^j - f^{j-1} \right\|_\infty = \max_k \left| f^j_{2k+1} - \left(f^{j-1}_k + \frac{f^{j-1}_{k+1} - f^{j-1}_k}{x^{j-1}_{k+1} - x^{j-1}_k} (x^j_{2k+1} - x^{j-1}_k) \right) \right| .$$

If the subdivision is only approximating, the even breakpoints have to be also taken into account in the computation of $\log \left\| f^j - f^{j-1} \right\|_\infty$.

5.1.2 Regularity of the limit function

This criterion has been proposed by F. Kuijt in [78] and is adapted here to fit non-uniform grids. The starting point is to recall the definition of the Hölder-regularity of a function.

Definition 13. *A function $f : I \subset \mathbb{R} \longrightarrow \mathbb{R}$ has $n + \alpha$ ($n \in \mathbb{N},\ 0 \leq \alpha \leq 1$) Hölder-regularity if*

$$\exists C < +\infty \text{ such that } \left| \frac{\partial^n f}{\partial x^n}(x_1) - \frac{\partial^n f}{\partial x^n}(x_2) \right| \leq C |x_1 - x_2|^\alpha,\ \forall (x_1, x_2) \in I^2.$$

Let us denote by $\rho^{[l]}_j = l! \max_k |{f^j_{k+1}}^{[l]} - {f^j_k}^{[l]}|$, and $\beta_j = \max_k |x^j_{k+1} - x^j_k|$, where ${f^j_k}^{[l]}$ denotes the usual l-th order finite difference at f^j_k. A subdivision scheme has $l + \alpha_l$ Hölder regularity if

$$\exists C < +\infty \text{ such that } \rho^{[l]}_j \leq C (\beta_j)^{\alpha_l}, \tag{5.2}$$

i.e. if the estimate of the l-th derivative converges uniformly with an exponent α_l. Assuming that the maximal value in equation (5.2) is reached, α_l can be approximated by

$$\alpha_l = \lim_{j \to +\infty} \frac{\log \frac{\rho^{[l]}_{j+1}}{\rho^{[l]}_j}}{\log \frac{\beta_{j+1}}{\beta_j}}. \tag{5.3}$$

The regularity of the limit function can be computed using equation (5.3). If the limit function has $n + \alpha_n$ regularity, the α_l should be equal to 1 for $l \in \{0, ..., n\}$, strictly smaller than 1 for $l = n + 1$, and be close to zero for $l > n + 1$.

5.1.3 Approximation order

Following the definition of approximation order given in Chapter 2 and using a reasoning similar to the two previous criteria, a criterion to estimate the approximation order of a scheme can be derived. Let g be a sufficiently smooth real-valued function. Let X_0 be the coarsest grid on which subdivision will be performed. The data $\{f^0_k\}$ living on X_0 is built using $f^0_k = g(x^0_k),\ \forall k$. Let us define also η as

$$\eta = \max_k \left| x^0_{k+1} - x^0_k \right| .$$

Let $\tilde{f}$ be the limit function obtained when $j \to \infty$. The *approximation order* is defined by the biggest integer p for which the following formula holds

$$\left\| \tilde{f} - g \right\|_\infty \leq C \eta^p, \text{with } 0 \leq C < +\infty.$$

Assuming that $\left\| f^j - g \right\|_\infty$ provides a good estimate for $\left\| \tilde{f} - g \right\|_\infty$, and that the upper bound is reached, let us define $\Delta_j(\eta) = \left\| f^j - g \right\|_\infty$ for a given η. Under the previous assumption, the following approximation holds

$$\Delta_j(\eta) \approx C\eta^p.$$

Therefore, we have

$$\log \Delta_j(\eta) \approx \log C + p \log \eta, \tag{5.4}$$

which implies that the $\log \Delta_j(\eta)$ behaves linearly with respect to $\log \eta$, with slope p providing an estimator of the approximation order.

5.2 Numerical criteria for surface subdivision schemes

In the previous section, we have studied several criteria to study the behavior of subdivision schemes on univariate data. A subset of these criteria can be adapted to surface subdivision. However, it is required for such criteria to be applicable to be able to compute the *distance* between two discrete surfaces. Such a measurement, based on the Hausdorff distance will be presented in Section 5.2.1. The adaptation of univariate criteria from Section 5.1 will be discussed in Section 5.2.2 for surfaces, and finally another numerical tool based on the study of the local curvature of the limit surface will be presented in Section 5.2.3.

5.2.1 Error metric for discrete surfaces

Computing the error, i.e. the distance between two discrete surfaces is not a trivial task, especially when there is neither an *a priori* knowledge of the surfaces, nor a one-to-one mapping between the vertices of the surfaces, which are both extremely restrictive conditions. In order to overcome the difficulties related to the type of data, the approach used to define such a distance metric will be based on the *Hausdorff distance* between two sets. We refer the reader to [3, 12] for more detailed studies. We will summarize the core concepts of such a metric in this section.

Let us first define the distance $d(p, \mathcal{S}')$ between a point p belonging to a surface $\mathcal{S}$ and a surface $\mathcal{S}'$ as

$$d(p, \mathcal{S}') = \min_{p' \in \mathcal{S}'} \left\| p - p' \right\|_2 . \tag{5.5}$$

From this definition, the *Hausdorff distance* between $\mathcal{S}$ and $\mathcal{S}'$, denoted by $d(\mathcal{S}, \mathcal{S}')$ is given by:

$$d(\mathcal{S}, \mathcal{S}') = \max_{p \in \mathcal{S}} d(p, \mathcal{S}'). \tag{5.6}$$

It is important to note that this distance is in general not symmetrical, i.e. $d(\mathcal{S}, \mathcal{S}') \neq d(\mathcal{S}', \mathcal{S})$. We will refer to $d(\mathcal{S}, \mathcal{S}')$ as *forward distance*, and to $d(\mathcal{S}', \mathcal{S})$ as *backward distance*. It is then convenient to introduce the *symmetrical Hausdorff distance* $d_s(\mathcal{S}, \mathcal{S}')$, defined as follows:

$$d_s(\mathcal{S}, \mathcal{S}') = \max \left[d(\mathcal{S}, \mathcal{S}'), d_s(\mathcal{S}, \mathcal{S}') = \max \left[d(\mathcal{S}, \mathcal{S}'), d(\mathcal{S}', \mathcal{S}) \right] . \tag{5.7}$$

The symmetrical distance provides a more accurate measurement of the error between two surfaces, since the computation of a "one-sided" error can lead to significantly underestimated distance values, as illustrated by Figure 5.1.

The point-to-surface distance defined in equation (5.5) can be used to define a *mean error* d_m between two surfaces $\mathcal{S}$ and $\mathcal{S}'$:

$$d_m(\mathcal{S}, \mathcal{S}') = \frac{1}{|\mathcal{S}|} \iint_{p \in \mathcal{S}} d(p, \mathcal{S}') dS, \tag{5.8}$$

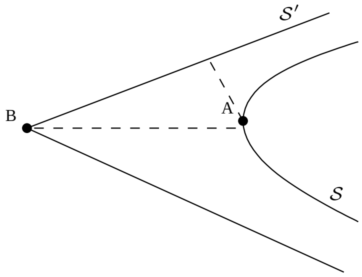


Figure 5.1: In this case, $d(\mathcal{S},\mathcal{S}')$ will remain much smaller than $d(\mathcal{S}',\mathcal{S})$, since here $d(A,\mathcal{S}') \ll d(B,\mathcal{S})$. Thus a small one-sided distance does not imply a small distortion.

where $|\mathcal{S}|$ denotes the area of $\mathcal{S}$. From this, the definition of a *root mean square error* d_{rmse} follows naturally :

$$d_{\text{rmse}}(\mathcal{S},\mathcal{S}') = \sqrt{\frac{1}{|\mathcal{S}|}\iint_{p\in\mathcal{S}} d(p,\mathcal{S}')^2 dS}. \tag{5.9}$$

Using equation (5.7), it is possible to define symmetrical versions of the mean and root-mean-square errors.

These tools provide ℓ_∞, ℓ_1 and ℓ_2 error metrics for discrete surfaces. While the point-to-surface distance defined in equation (5.5) can be computed analytically (since it is nothing more than a “point-to-triangle” distance), computing the Hausdorff distance from equation (5.6) require to sample $\mathcal{S}$ in order to achieve sufficient precision. The implementation detailed in [3] has been used to compute distortions between surfaces.

5.2.2 Adaptation of univariate criteria

Using the error metric defined in the previous section, the adaptation of the convergence criterion proposed in Section 5.1.1 is rather straightforward. Let us denote by $\mathcal{S}^j$ the triangulated surface obtained after j levels of subdivision. The surface subdivision scheme converges if there exists $\beta > 0$ and $0 < \alpha < 1$ such that for all j

$$d_s(\mathcal{S}^j,\mathcal{S}^{j-1}) < \beta\alpha^{j-1}.$$

As in the univariate case, taking the logarithm leads to a linear relation between $\log d_s(\mathcal{S}^j,\mathcal{S}^{j-1})$ and the level index j, which provides the upper bound for the convergence parameter.

However, the regularity and approximation order estimator are not so easily adapted from the univariate case, since they require to have a knowledge of a parameterization of the surface throughout the levels. While this assumption can be satisfied for simple analytic surfaces, e.g. a torus, having such a parameterization for any surface is a non-trivial task (although recent works on the subject are promising [29, 56, 71]).

5.2.3 Curvature

As suggested in [111] and [89], *curvature* is a characteristic to take into account when comparing subdivision schemes, for instance if the curvature remains bounded. Let us first remind the main concepts regarding the curvature of a continuous surface, following the notations of Section 3.1. More details regarding this topic can be found in differential geometry books [77, 98].

Let us consider a surface, described by a function f as $z = f(x,y)$, such that the graph of f is tangent to the plane (x,y) at the origin (which is always possible, at least locally, as stated by the implicit function theorem). The *second fundamental form* of f at the origin is defined by

$$\mathrm{II} = D^2 f = \begin{pmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial x \partial y} & \frac{\partial^2 f}{\partial y^2} \end{pmatrix}.$$

The *principal curvatures* of the surface, denoted by κ_1 and κ_2 are the eigenvalues of this second fundamental form (we assume that it is not degenerate at the origin), and the *principal curvature directions* are given by the eigenvectors associated to these eigenvalues, i.e. in the appropriate coordinate system, the second fundamental form at the origin becomes

$$\mathrm{II} = \begin{pmatrix} \kappa_1 & 0 \\ 0 & \kappa_2 \end{pmatrix}.$$

Using this coordinate system, the *mean curvature* H is defined by

$$H = \frac{1}{2}\mathrm{Tr}\ \mathrm{II} = \frac{\kappa_1 + \kappa_2}{2},$$

and the *Gaussian curvature* G is defined by

$$G = \det \mathrm{II} = \kappa_1 \kappa_2.$$

The curvature of discrete surfaces can be estimated, for instance using the method described in [95]. This technique relies on the *Gauss-Bonnet* theorem.

Theorem 20. *(Gauss-Bonnet) Let us consider a smooth disk D in a two-dimensional manifold. Then*

$$\int_D G + \int_{\partial D} \kappa_g = 2\pi,$$

where κ_g denotes the geodesic curvature of the boundary ∂D. If ∂D has corners with interior angles α_i, the above relation becomes*

$$\int_D G + \int_{\partial D} \kappa_g + \sum_i (\pi - \alpha_i) = 2\pi.$$

Using this theorem on the neighborhood of a vertex provides a reliable way of estimating the Gaussian curvature around a vertex. The mean curvature can also be estimated by taking into account that the mean normal curvature is also the Laplace-Beltrami operator which generalizes the Laplace operator to manifolds. Since the mean curvature is computed from the norm of the *mean curvature vector* $\boldsymbol{H}$, using the relation

$$\boldsymbol{H} = H\boldsymbol{n},$$

where $\boldsymbol{n}$ denotes the normal vector to the surface, the sign of H is clearly not known, since both $\boldsymbol{n}$ and $-\boldsymbol{n}$ are admissible normal vectors for the considered surface. Therefore, in our experiments, we will focus on the evolution of the maximum of $|H|$. The Gaussian curvature, given its intrinsic nature (cf. Gauss's *Theorema Egregium*) does not suffer from similar problems. Therefore, our experiments will study the evolution of both minimum and maximum values of the Gaussian curvature.

5.3 Results for univariate schemes

In this section, we detail the results achieved using the numerical criteria proposed in Section 5.1. A visual comparison of the limit curves generated will also be proposed.

---

*The geodesic curvature at point P of a curve $\mathcal{C}$ lying on a surface $\mathcal{S}$ is the curvature of the orthognal projection of $\mathcal{C}$ on the plane tangent to $\mathcal{S}$ at P.

5.3.1 Tests performed

The proposed univariate scheme, as described in Section 4.2, is compared with the four-point scheme defined in equation (2.8) (or equation (2.9) with $w = \frac{1}{16}$) when using uniform initial grids. For non-uniform grids, the adaptation described in equation (2.10) will be used. These "reference" schemes have been chosen because of the support of their mask, and because of their interpolating nature. The linear schemes used as reference lead to $\mathcal{C}^{2-\varepsilon}$ limit functions, and have an approximation order equal to 4, since they reproduce cubic polynomials (cf. Section 2.3.1).

The grids are generated by sampling $\mathcal{C}^\infty$ functions, one Gaussian f

$$f(x) = e^{-x^2},$$

and one polynomial

$$p(x) = x^2 + \frac{x^4}{1000}.$$

In the non-uniform case, the initial grid used is given in Table 5.1.

k	0	1	2	3	4	5	6	7	8	9
x_k^0	-2	-1.5	-1.4	-1	0	0.2	0.6	0.9	1.3	2

Table 5.1: Non-uniform initial sampling grid

Another non-uniform initial grid, shown in Table 5.2 will be considered.

k	0	1	2	3	4	5	6	7	8
x_k^0	-2	-1	0	1	4	5	7	9	11
f_k^0	8	5.5	4	3	$\frac{1}{4}$	1	3	5.5	8

Table 5.2: Non-uniform initial grid

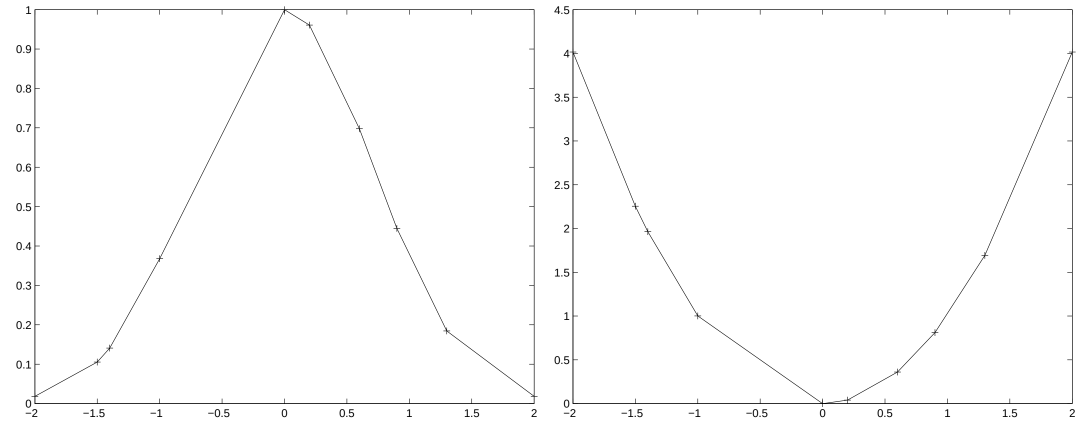
The aspect of the non-uniform grids is shown in Figure 5.2.

The various criteria defined in Section 5.1 are computed for all grids and all schemes. In addition, when the initial grid is sampled from a function, the ℓ_2 error between the piecewise linear function obtained through subdivision and the original function will be computed.

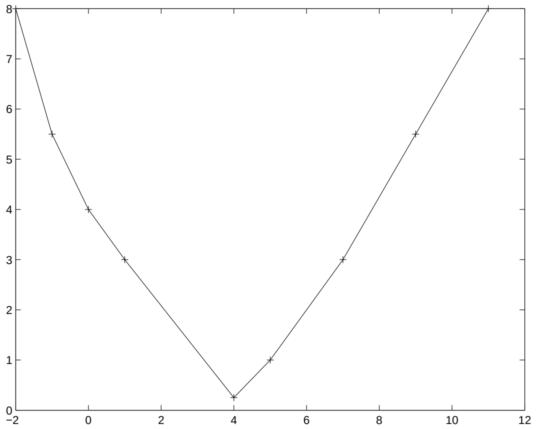
5.3.2 Convergence and regularity

In this section, we compare the convergence parameters from equation (5.1), as well as the α_l of equation (5.3).

Let us first consider a regular grid over the interval $[-2; 2]$, on which the proposed functions f and p are sampled. The tests have been performed using 5, 10 and 15 samples in the initial grid, and 17 levels of subdivision. The convergence parameters achieved using regular grids are summarized in Table 5.3. The "Spherical (or.)" label in the table refers to the spherical subdivision scheme using the function h from equation (4.18), and conversely, the label "Spherical (alt.)" refers to the spherical scheme with function h defined by equation (4.32).



(a) f sampled over the grid from Table 5.1 **(b)** p sampled over the grid from Table 5.1



(c) Grid from Table 5.2

Figure 5.2: Non-uniform initial grids

		β	α
f	4-point	0.174611	0.294038
	Spherical (or.)	0.194554	0.292088
	Spherical (alt.)	0.194637	0.293585
p	4-point	0.344173	0.284295
	Spherical (or.)	0.332720	0.279352
	Spherical (alt.)	0.347747	0.286311

(a) 5 samples in initial grid

		β	α
f	4-point	0.043562	0.273818
	Spherical (or.)	0.043226	0.274491
	Spherical (alt.)	0.043226	0.274491
p	4-point	0.068505	0.284306
	Spherical (or.)	0.063057	0.280043
	Spherical (alt.)	0.063057	0.280043

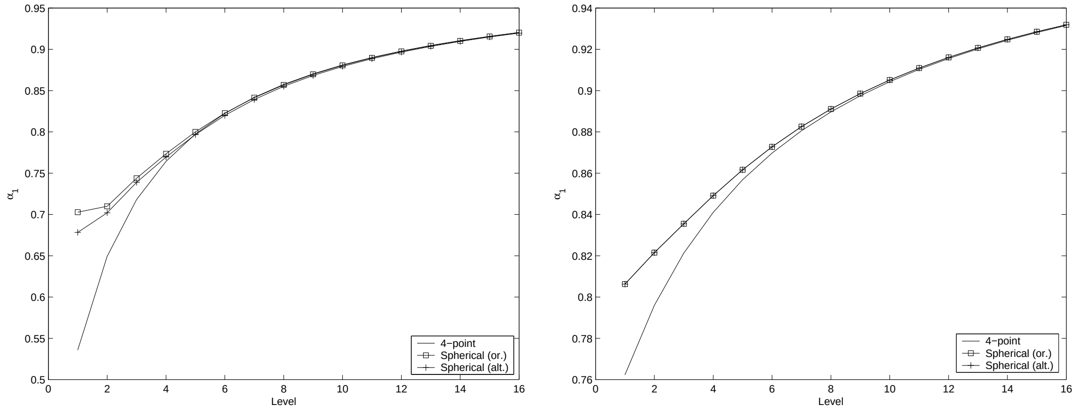
(b) 10 samples in initial grid

		β	α
f	4-point	0.019443	0.267235
	Spherical (or.)	0.020288	0.271342
	Spherical (alt.)	0.020288	0.271342
p	4-point	0.028395	0.284288
	Spherical (or.)	0.027047	0.282347
	Spherical (alt.)	0.027047	0.282347

(c) 15 samples in initial grid

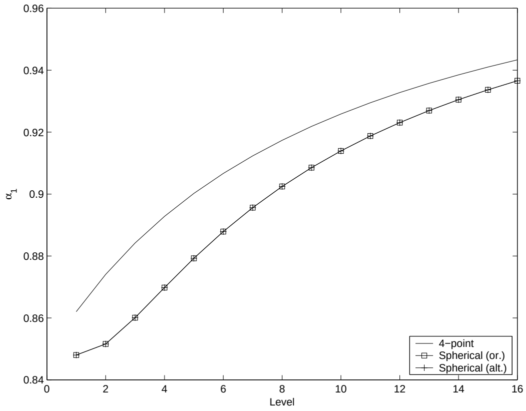
Table 5.3: Convergence parameters - regular initial grid

As suspected, the parameter α_0 remains almost constant and equal to 1, and α_2 close to 0. The evolution of parameter α_1 (which implies that the Hölder regularity of the limit function will be at least $1 + \alpha_1$) versus the level of subdivision is shown in Figure 5.3 for grids sampled from f and Figure 5.4 for grids sampled from p.



(a) 5 samples in initial grid

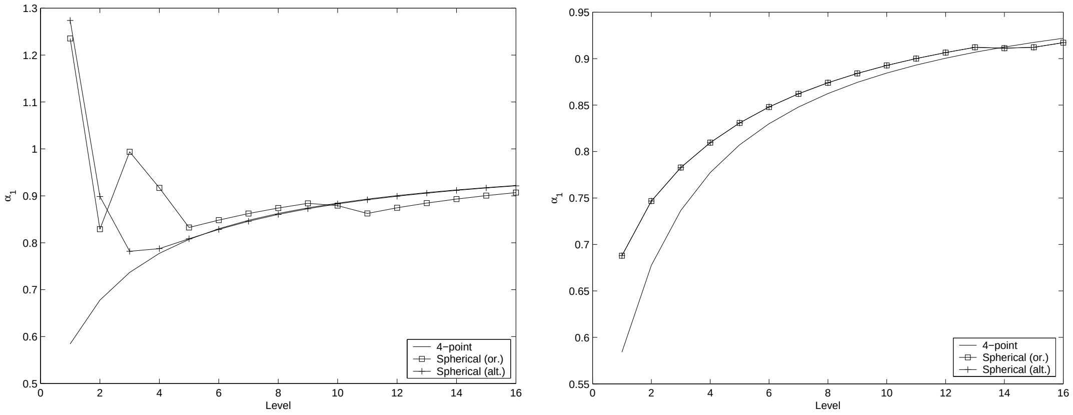
(b) 10 samples in initial grid



(c) 15 samples in initial grid

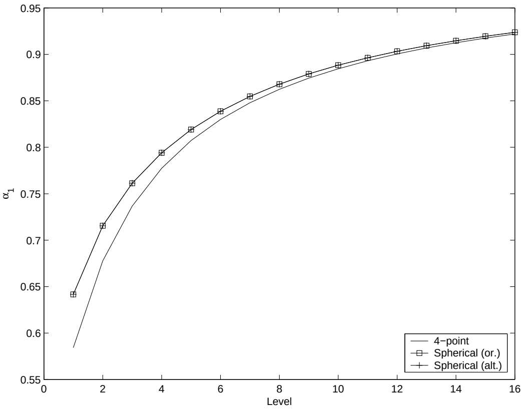
Figure 5.3: Evolution of α_1 vs. subdivision level - initial grid sampled with f

The convergence parameters achieved using the irregular initial grids from Table 5.1 and Table 5.2 are presented in Table 5.4. The evolution of α_1 versus the level of subdivision is shown in Figures 5.5 and 5.6. These results will be commented in Section 5.3.5.



(a) 5 samples in initial grid

(b) 10 samples in initial grid



(c) 15 samples in initial grid

Figure 5.4: Evolution of α_1 vs. subdivision level - initial grid sampled with p

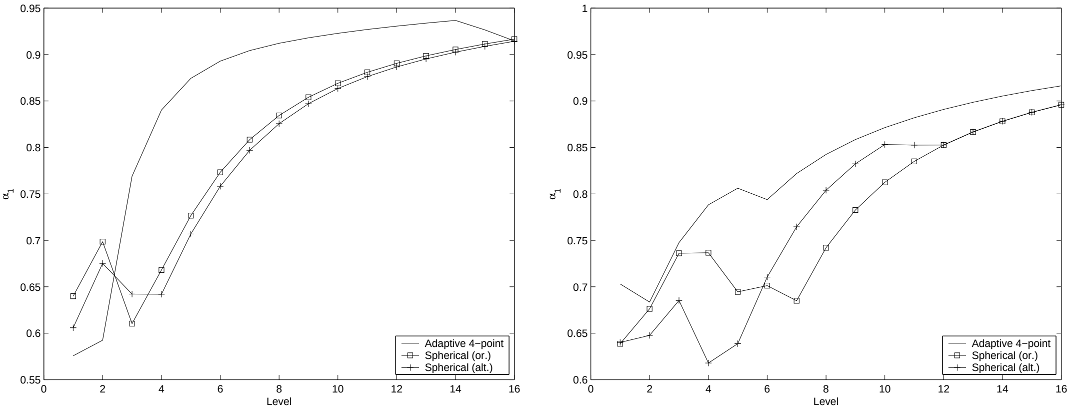
		β	α
f	Adaptive 4-point	0.191911	0.273067
	Spherical (or.)	0.037903	0.287306
	Spherical (alt.)	0.038950	0.286946
p	Adaptive 4-point	0.239559	0.277674
	Spherical (or.)	0.125560	0.280021
	Spherical (alt.)	0.125560	0.280021

(a) Initial irregular grid from Table 5.1

	β	α
Adaptive 4-point	1.106508	0.283758
Spherical (or.)	0.458076	0.293626
Spherical (alt.)	0.500810	0.301783

(b) Initial irregular grid from Table 5.2

Table 5.4: Convergence parameters - irregular initial grids



(a) Initial grid sampled with f **(b)** Initial grid sampled with p

Figure 5.5: Evolution of α_1 vs. subdivision level - initial grid from Table 5.1 used.

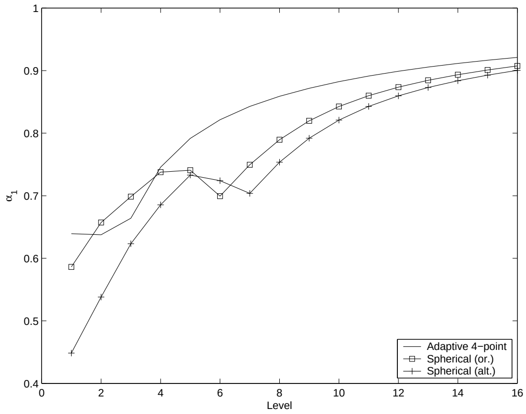


Figure 5.6: Evolution of α_1 vs. subdivision level - initial grid from Table 5.1 used.

5.3.3 Approximation order

In order to estimate the approximation order of subdivision schemes using equation (5.4), we will consider regular initial grids, over which values of f and p are taken to define the coarsest grid. The functions are sampled over the interval $[-5; 5]$ and the parameter η varies between 1 and 10^{-3}. Each grid has been subdivided 15 times. An example of the evolution of $\log \Delta_{15}$ versus $\log \eta$ is shown in Figure 5.7. Figure 5.7 clearly shows a linear part followed by a saturation when η becomes small enough. Therefore, the approximation order will be estimated using the non-saturated part of the plot. The estimates of approximation order are summarized in Table 5.5.

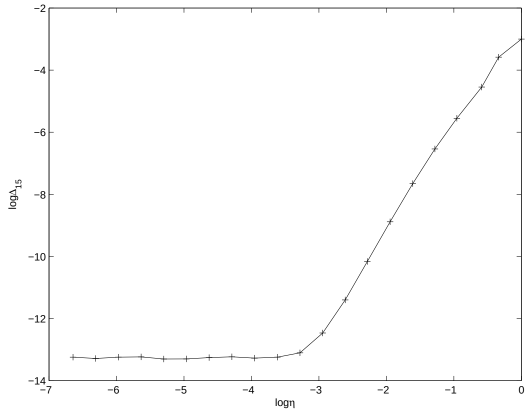
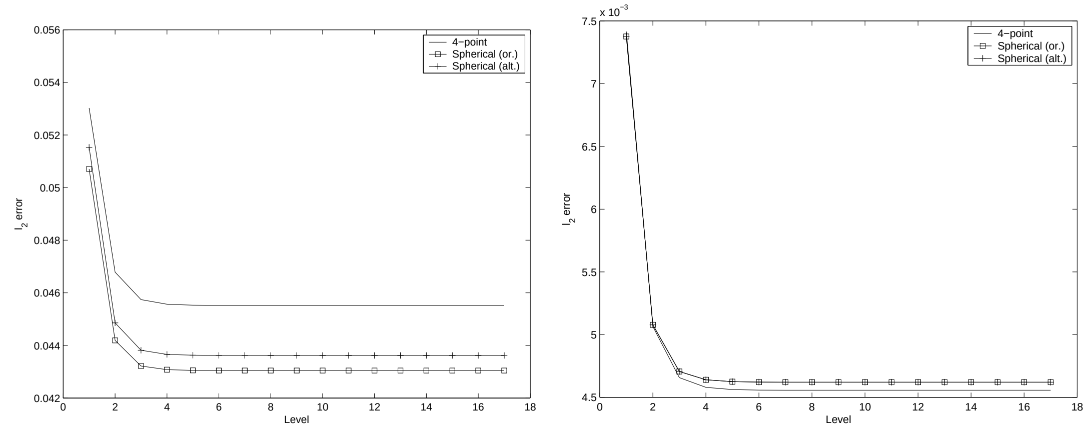


Figure 5.7: Evolution of $\log \Delta_{15}$ vs. $\log \eta$

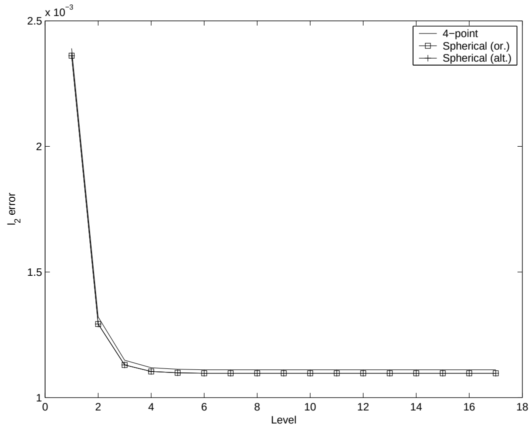
		Approximation order
	4-point	3.4076
f	Spherical (or.)	3.2634
	Spherical (alt.)	3.2635
	4-point	1.9326
p	Spherical (or.)	1.9324
	Spherical (alt.)	1.9324

Table 5.5: Approximation order estimates

In addition to the approximation order estimates, we also plot the evolution of the ℓ_2-error between the piecewise linear function obtained through subdivision and the function used to compute the initial grid. Figure 5.8 presents the results achieved with f and Figure 5.9 those obtained using p as reference, both using uniform starting grids. The results using non-uniform initial grid from Table 5.1 are shown in Figure 5.10.

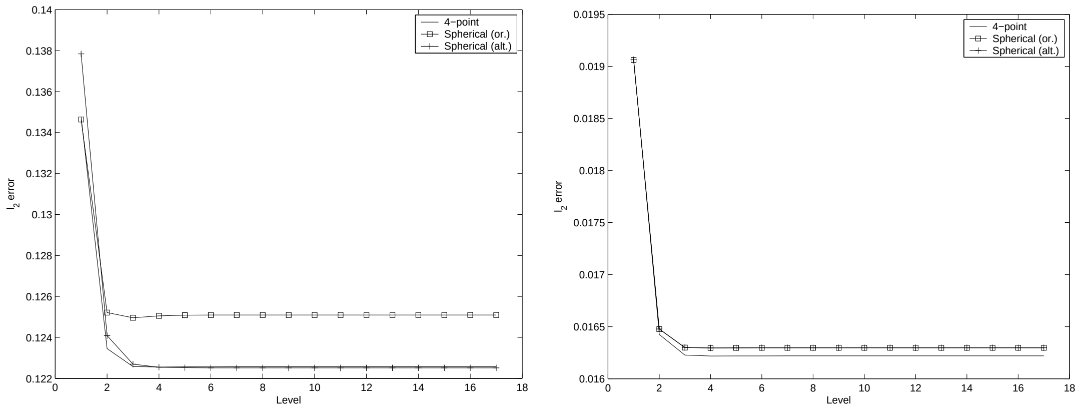


(a) 5 samples in initial grid **(b)** 10 samples in initial grid

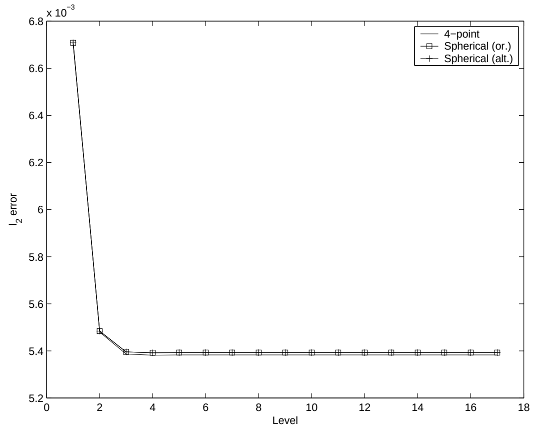


(c) 15 samples in initial grid

Figure 5.8: ℓ_2-error between f and the approximation obtained through subdivision. Initial regular grid sampled using f



(a) 5 samples in initial grid **(b)** 10 samples in initial grid



(c) 15 samples in initial grid

Figure 5.9: ℓ_2-error between p and the approximation obtained through subdivision. Initial regular grid sampled using p

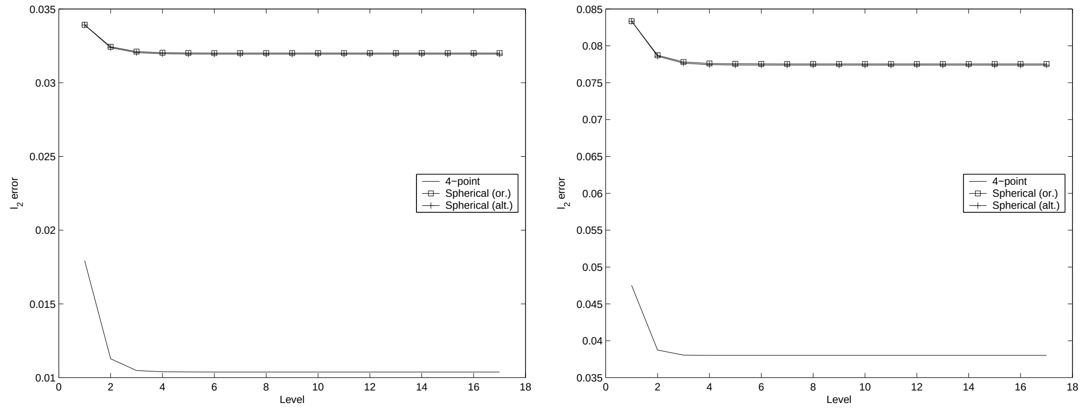
5.3.4 Visual comparison

In order to complete the comparison of univariate scheme, we present here typical limit curves obtained using the proposed non-linear scheme and the four-point one. Figure 5.11 shows the function obtained (after 10 levels of subdivision), using a "unit pulse" as initial grid. Finally, the adaptation of the spherical scheme for planar polygons is compared to the 4-point scheme in Figure 5.12.

5.3.5 Analysis

Let us now analyze the numerical results presented in the previous sections.

The convergence parameters estimated for the different types of grids and presented in Table 5.3 and Table 5.4, are consistent with the fact that all proposed schemes converge. The results from



(a) Initial grid sampled using f

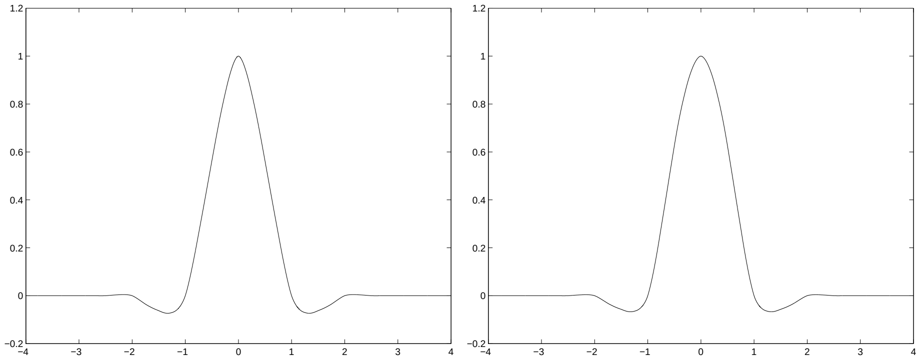
(b) Initial grid sampled using p

Figure 5.10: ℓ_2-error between the approximation obtained through subdivision. Initial non-uniform grid from Table 5.1 used.

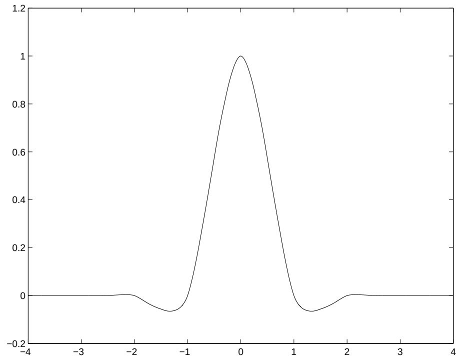
the mentioned tables show a very close behavior between the 4-point scheme and the proposed non-linear scheme. The influence of the choice of the function h for the non-linear scheme is not really visible in these results. Even with non-uniform initial grids, the α parameter does not exhibit strong differences between the adaptive four-point scheme and the spherical subdivision scheme. Recalling the proof of Theorem 8, the convergence rate α is given by $\left\| \left(\frac{1}{n}S_b\right)^l \right\|_\infty^{1/l}$. In the case of the four-point scheme, as shown in Section 2.3.3, this quantity is equal to $\frac{1}{2}+2|w|$, i.e. for the case $w=\frac{1}{16}$, we have $\alpha=\frac{5}{8}=0.625$. Similarly, as presented in Section 4.3, the convergence rate for spherical subdivision schemes was at most $\frac{\sqrt{1+\sin^2\theta_0}}{2}$, i.e. close to 0.55 or 0.52 depending on the choice of h. Since our estimates of α exhibit values roughly between 0.26 and 0.3, which seems to prove that more critical cases could be imagined to obtain convergence rate estimates that are closer to the theoretical ones.

The approximation order estimates presented in Table 5.5 also show that the behavior of the 4-point and spherical scheme are very close. Given that the 4-point scheme has an approximation order equal to 4, our estimates are clearly pessimistic, especially for the grid sampled from a polynomial. The x^4 term avoids to fall into the class of polynomials that are reproduced by the 4-point scheme, but even with a small weight, the estimate of approximation order in this case falls down to 2. Anyway, the close behavior of both schemes in this field is also a positive sign. The evolution of the ℓ_2-error between the subdivided and the reference function, presented in Figure 5.8, Figure 5.9 and Figure 5.10 show that the schemes behave very closely, although the spherical scheme seems to be better when using a very coarse initial grid. In the irregular setting, the spherical scheme is outperformed by the adaptive 4-point scheme, but remains within acceptable error ranges. The tests performed do not show a clear influence of h on the results.

The regularity of the limit function again shows close behaviors between the 4-point scheme and the spherical scheme. As seen in Chapter 3, the 4-point scheme leads to $\mathcal{C}^{2-\varepsilon}$ limit functions, which is confirmed by the evolution of α_1 shown in Figures 5.3, 5.4, 5.5 and 5.6. Since α_1 gets closer to 1 as the level of subdivision increases, the limit function has an estimated Hölder regularity that gets closer to 2, which is consistent with the fact that limit functions obtained using the 4-point scheme are $\mathcal{C}^{2-\varepsilon}$. In some cases, the limit function obtained using spherical subdivision can be smoother that the one obtained using the 4-point scheme, but the difference achieved is not really



(a) Linear 4-point scheme **(b)** Spherical (or.) scheme

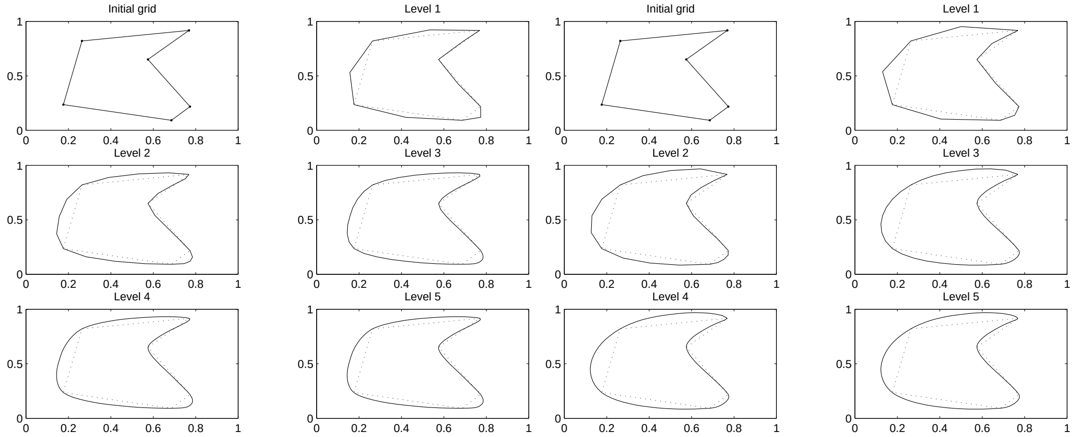


(c) Spherical (alt.) scheme

Figure 5.11: Limit function for unit pulse (10 levels of subdivision)

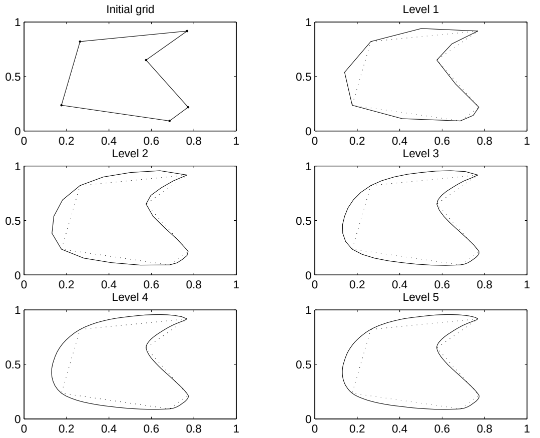
significant. In the irregular setting, the adaptive 4-point scheme is better than the spherical one, although the difference decreases with the subdivision level. The influence of h is also not really clear here, although a small advantage for the one defined in equation (4.18) is visible in some cases.

The visual comparison proposed confirms the trends given by the numerical estimates, i.e. that both linear and non-linear schemes have a close behavior and lead to visually pleasant limit curves, as shown in Figure 5.11 or Figure 5.12. Finally, about the respective complexity of the methods, the non-linear one is clearly more complex than the linear ones, given the sines and cosines that need to be computed at each step. However, computation times remain close between linear and non-linear methods.



(a) Linear 4-point scheme

(b) Spherical (or.) scheme



(c) Spherical (alt.) scheme

Figure 5.12: Comparison of planar polygon subdivision (5 levels)

5.4 Results for surface subdivision schemes

In this section we present the results achieved using the surface subdivision scheme proposed in Chapter 4. A visual comparison between the limit surfaces achieved will also be proposed.

5.4.1 Tests performed

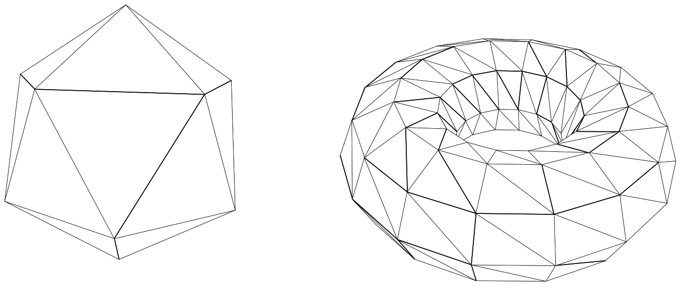
The tests described in Section 5.2 are used, as in the univariate case to derive the convergence parameters for the proposed scheme. It will be compared to the Butterfly scheme, as described in [145], and to the Loop scheme as described in [88]. Although the Loop scheme is not interpolating, but only approximating, it provides a good reference in terms of convergence and regularity of the limit surface.

5.4.2 Convergence

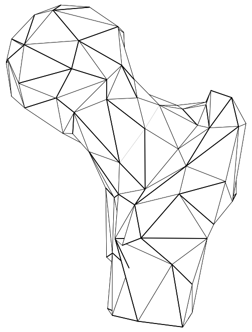
According to the method used to estimate the convergence parameters, there are no restrictions on the initial surface used to perform the test. In our tests, we have chosen to use 3 different initial surfaces

- an icosahedron,
- a coarse torus,
- a coarse version of the "Balljoint" model.

The initial torus has 128 triangles, and the coarse "Balljoint" model has 200 triangles. Given the complexity of the computations of distortions between very large models and the exponential growth of the number of triangles, the torus and the Balljoint will only be subdivided six times, and the icosahedron seven times. When available (i.e. for the torus and sphere), the $\ell_2$and ℓ_∞ errors between the subdivided model and the reference model will be computed. The initial coarse models are shown in Figure 5.13. The convergence parameters estimated using these models are presented in Table 5.6.



(a) Icosahedron **(b)** Torus



(c) Balljoint

Figure 5.13: Initial coarse models. The "Balljoint" original model has been simplified using `QSlim` [53] to produce this coarse version

	α	β
Butterfly	0.35331	3.2825
Spherical (or.)	0.2880	5.7519
Spherical (alt.)	0.2846	4.9341
Loop	0.2995	6.6585

(a) Icosahedron

	α	β
Butterfly	0.2766	1.3354
Spherical (or.)	0.2804	1.3460
Spherical (alt.)	0.2804	1.3459
Loop	0.2507	1.5383

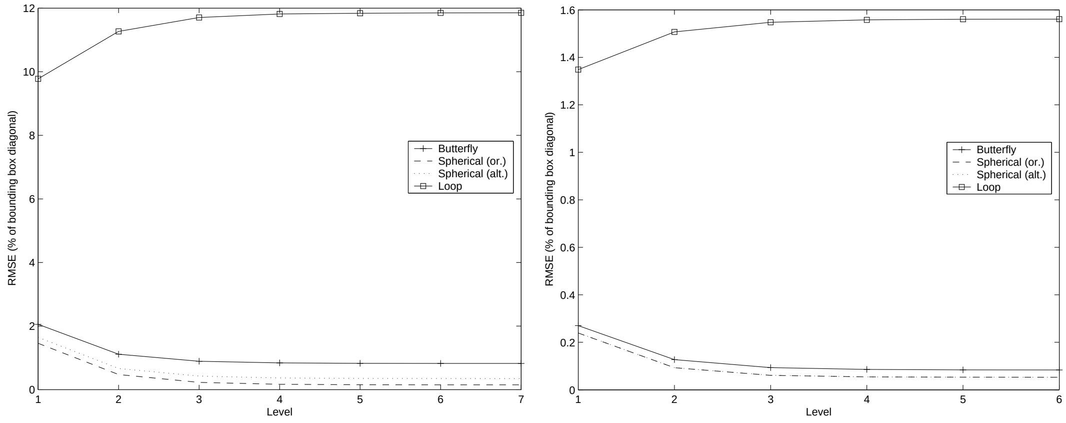
(b) Torus

	α	β
Butterfly	0.3356	3.5105
Spherical (or.)	0.3610	2.4458
Spherical (alt.)	0.3857	2.1805
Loop	0.2886	3.1332

(c) Balljoint

Table 5.6: Convergence parameters

The evolution of the ℓ_2 error between the analytic surface and the subdivided one is shown in Figure 5.14. Figure 5.15 shows the evolution of the ℓ_∞ errors under the same conditions.

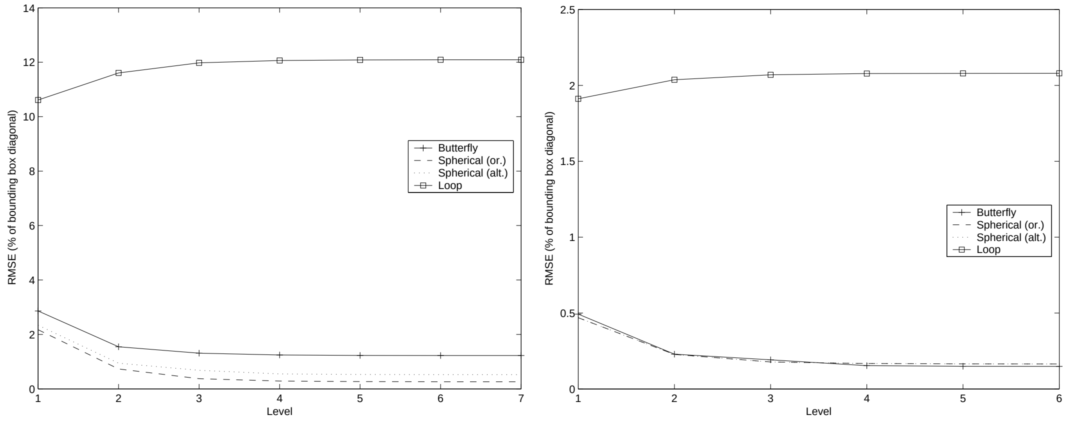


(a) Icosahedron **(b)** Torus

Figure 5.14: Evolution of the ℓ_2 error between the subdivided model and the analytic reference surface.

5.4.3 Curvature

In this section, we present the evolution of the mean and Gaussian curvatures. Figure 5.16 presents the evolution of curvature when using the icosahedron as initial model, and Figure 5.17 presents results achieved when using the coarse torus as initial model. Figure 5.18 presents the results for the “Balljoint” model.



(a) Icosahedron

(b) Torus

Figure 5.15: Evolution of the ℓ_∞ error between the subdivided model and the analytic reference surface.

5.4.4 Visual comparison

In this section we provide a visual comparison between the different schemes. Figure 5.19 shows the icosahedron subdivided 4 times with each considered subdivision scheme. The Balljoint model, after 3 levels is shown in Figure 5.20. Finally, an example of mesh having a boundary is shown in Figure 5.21.

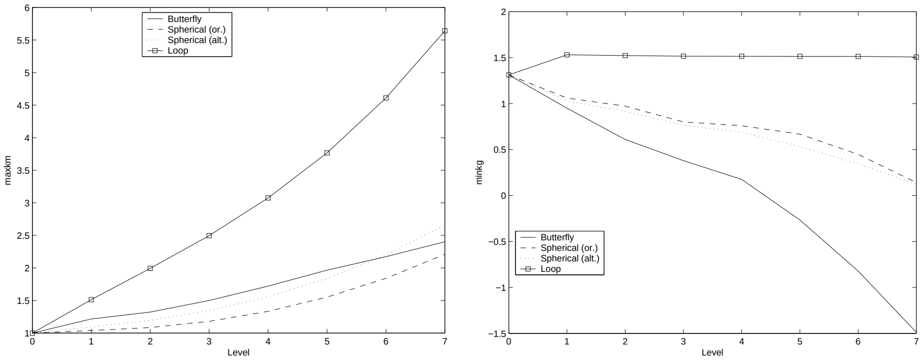
5.4.5 Analysis

Let us now analyze the numerical results given in the previous sections.

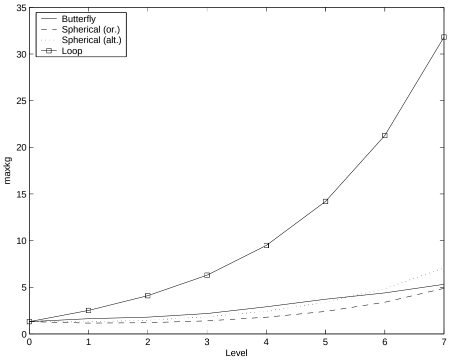
The convergence parameters given in Table 5.6 show that the convergence rates are close one to another. Loop's scheme seems to have a small advantage in some cases, but remain close to the other schemes. The influence of h on the convergence rate remains unclear, in one case the convergence rate is (very slightly) better, in another one it is identical and in the last one it is (slightly) worse.

The evolution of the ℓ_2 and ℓ_∞ errors from Figures 5.14 and 5.15 show that the spherical scheme approximates better the analytic reference surface, which is not too surprising, given the method used, when dealing with a icosahedron or a torus. Not surprisingly, the Loop scheme, given its approximating nature, gives error values much higher than the interpolating schemes.

The evolution of curvatures is quite interesting. Let us first remind the theoretical values for the reference surfaces we consider here, i.e. the unit sphere and a torus. A sphere of radius r has a Gaussian curvature equal to 1 and a mean curvature equal to $\frac{1}{r}$. In our tests, the icosahedron is sampled from a sphere of radius $r = 1$. It is interesting to note that the precision of the numerical method used to compute the curvatures is roughly 1% for a sphere tessellated with 327680 triangles. As shown in Figure 5.16, the spherical scheme is the one that offers the best compromise in terms of curvature. The increasing mean curvature shown by the Loop scheme could be partially caused by the well-known “shrinking” phenomenon of this scheme, but not entirely. Since this trend is also confirmed by the Gaussian curvature, one can suspect the existence of singular points on the limit surface around which the curvature is high. In the case of a torus of radii R and r (r being the



(a) $\max|H|$ **(b)** $\min G$



(c) $\max G$

Figure 5.16: Evolution of curvatures vs. subdivision level - Icosahedron

radius of the "tube"), the mean curvature of the torus is

$$H_{\text{torus}} = \frac{1}{2r}\left(2 - \frac{1}{1+\frac{r}{R}\cos\phi}\right),$$

which implies that the maximal value of this quantity is equal to

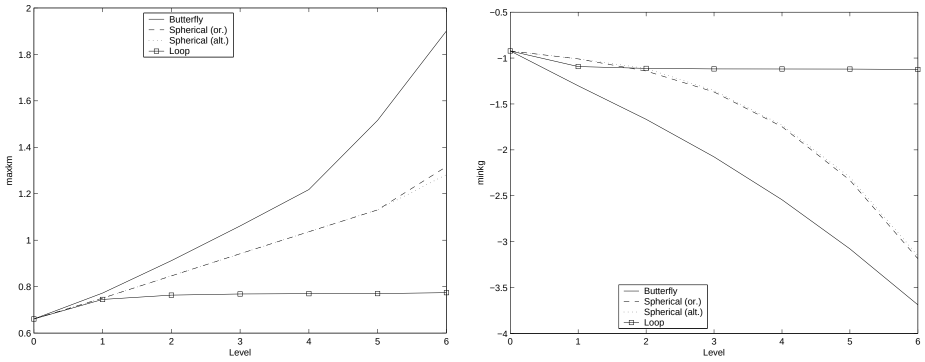
$$\max|H_{\text{torus}}| = \frac{1}{2r}\left(2 - \frac{1}{1-\frac{r}{R}}\right).$$

In our tests, we have used a torus having $R = 2$ and $r = 1$, which leads to $\max|H_{\text{torus}}| = \frac{2}{3}$. Similarly, the Gaussian curvature of a torus is

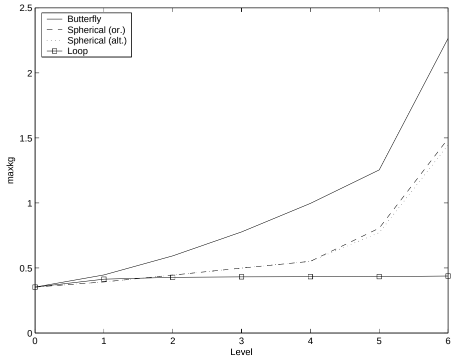
$$G_{\text{torus}} = \frac{1}{r^2}\left(1 - \frac{1}{1+\frac{r}{R}\cos\phi}\right).$$

Therefore, the maximal and minimal values of the Gaussian curvature of a torus are given by

$$\begin{aligned}
\min G_{\text{torus}} &= \frac{1}{r^2}\left(1 - \frac{1}{1+\frac{r}{R}}\right), \\
\max G_{\text{torus}} &= \frac{1}{r^2}\left(1 - \frac{1}{1-\frac{r}{R}}\right).
\end{aligned}$$



(a) $\max |H|$ **(b)** $\min G$

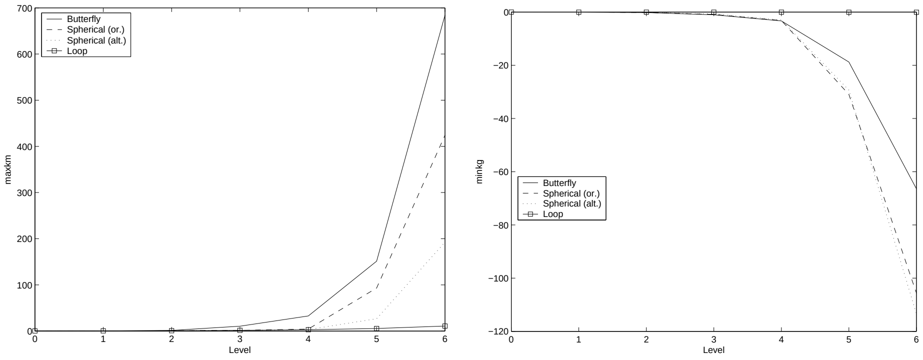


(c) $\max G$

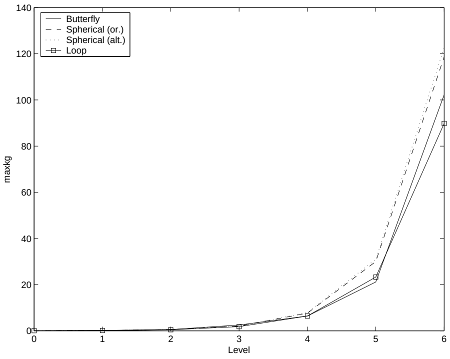
Figure 5.17: Evolution of curvatures vs. subdivision level - Torus

For the values of r and R considered in the test, we finally have $\min G_{\text{torus}} = -1$ and $\max G_{\text{torus}} = \frac{1}{3}$. As shown by Figure 5.17, Loop's scheme is the one giving the best results in terms of curvature preservation. The Butterfly gives the worst results, and the spherical scheme, although following the trend of diverging curvature shown by the Butterfly gives better results. Finally, the same experiment conducted for the Balljoint model lead to the results shown in Figure 5.18. While the spherical scheme is better for the mean curvature, the Butterfly scheme leads to better results when looking at the Gaussian curvature. In either case, the Loop scheme leads to better results for this model. However, while the mean and maximal Gaussian curvatures seem to be preserved by Loop's scheme, the minimal Gaussian curvature has a tendency to diverge, which also seems to imply that a singularity exists in the Loop scheme's limit surface. This overall trend of diverging curvature for the interpolating schemes is most likely related to the fact that the initial coarse model contains very sharp transitions and peaks. Here again, the influence of h does not appear clearly.

The visual comparison of Figure 5.19 shows an advantage in favor of the spherical scheme, which given the method is not too surprising. When looking at Figure 5.20, the Loop scheme clearly leads to a smoother limit surface. The limit surface generated by the Butterfly scheme shows many ripples (probably linked to the presence of irregular vertices). The models obtained using the spherical scheme, although having "patches", which are more linked to irregularities in the triangulation than surface discontinuities, give an interesting compromise between Butterfly and Loop. The results of



(a) $\max|H|$ **(b)** $\min G$



(c) $\max G$

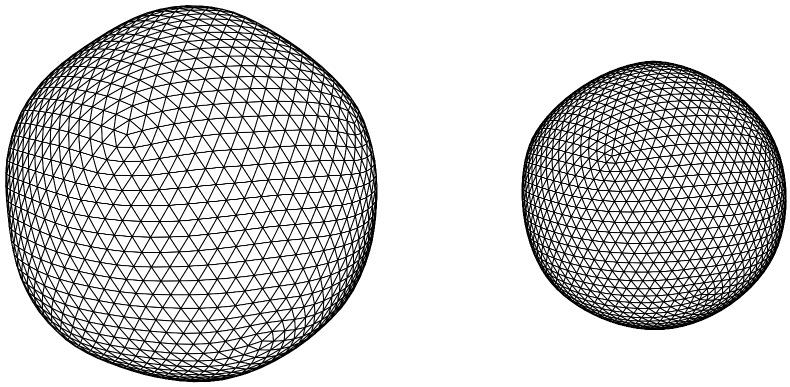
Figure 5.18: Evolution of curvatures vs. subdivision level - Balljoint

Figure 5.21 also shows that the adaptation of the spherical scheme is clearly suitable for meshes having boundaries.

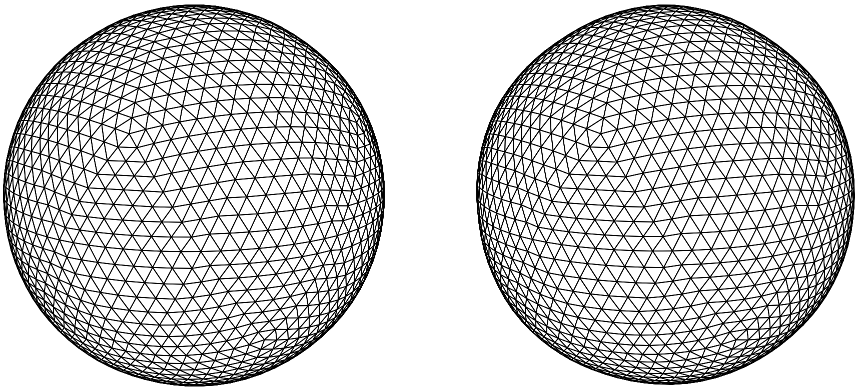
Finally regarding the respective complexity of the techniques, the Loop and Butterfly are identical. The spherical scheme is by construction more complex, but remains quite efficient. As an example, a model having 82000 triangles is subdivided in 0.02 seconds using the Butterfly scheme and 0.04 seconds using the spherical scheme (on a Pentium III running at 666 MHz). Therefore, the increased complexity of the scheme is still acceptable for most applications.

5.5 Conclusion

In this chapter, we have proposed a number of numerical methods to estimate the main parameters of a subdivision scheme. When applied to known linear schemes, those estimates provided values consistent with the theoretical ones. The spherical subdivision scheme proposed in Chapter 4 has a behavior close to the 4-point/Butterfly scheme, sometimes better for specific features such as curvature preservation. While being far from proving theoretically the behavior of the spherical scheme, those estimates give us an intuitive view of what can be expected from this non-linear scheme. Several questions remain open regarding the theoretical studies of the scheme, however, those estimates provide a sound basis to investigate further this domain.

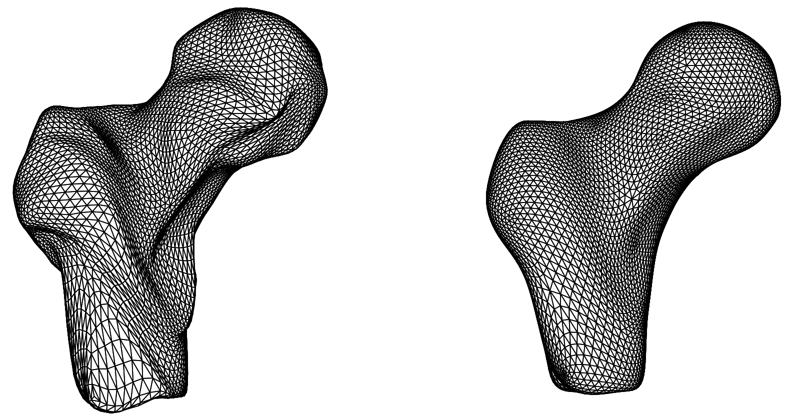


(a) Butterfly **(b)** Loop

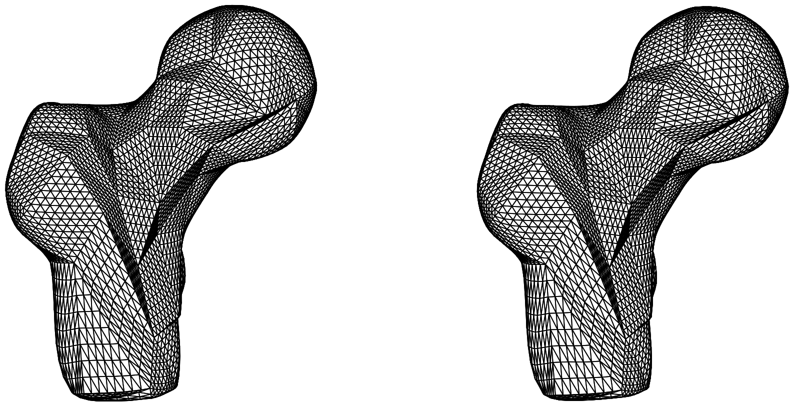


(c) Spherical (or.) **(d)** Spherical (alt.)

Figure 5.19: Icosahedron subdivided 4 times

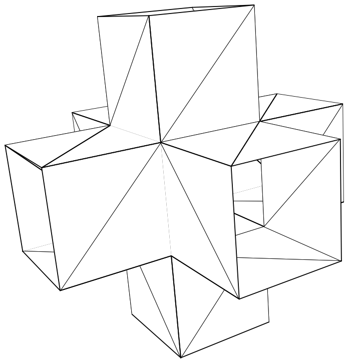


(a) Butterfly **(b)** Loop

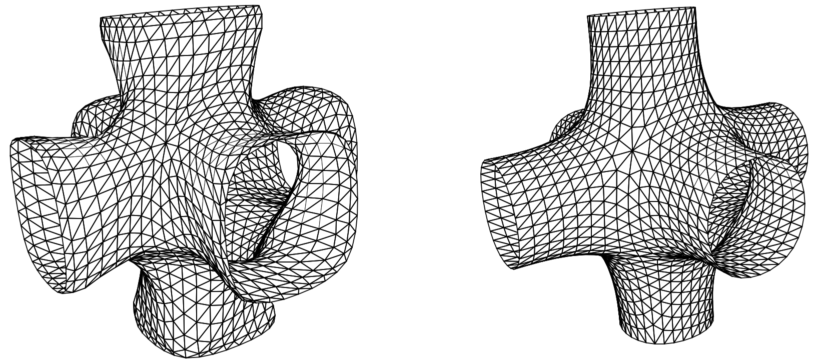


(c) Spherical (or.) **(d)** Spherical (alt.)

Figure 5.20: Balljoint subdivided 3 times

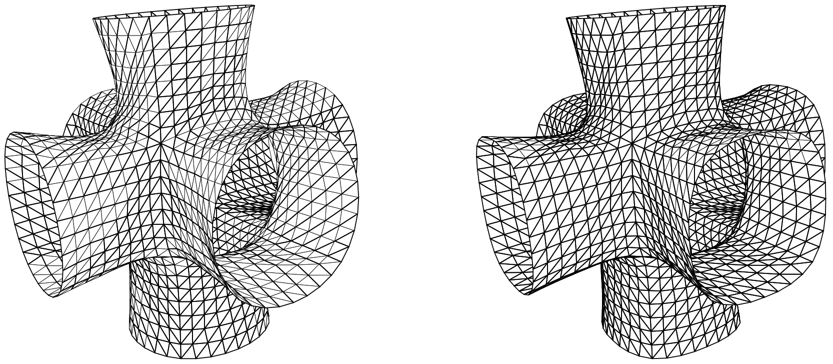


(a) Initial mesh



(b) Butterfly

(c) Loop



(d) Spherical (or.)

(e) Spherical (alt.)

Figure 5.21: "Pipes" model subdivided 3 times

Application to 3D model compression

6

There are several classes of applications for subdivision. In the case of surfaces represented by polygonal (triangular) meshes, as mentioned in Chapter 3, an application for computer-aided design could be to generate quickly smooth surfaces, having a shape controllable by only few control points. Another idea, as suggested in Section 2.3.1, is based on the close relationship between multiresolution analysis and subdivision, which naturally leads to the idea of building multiresolution representations of discrete surfaces. This idea is quite similar to the concept of multiresolution representation of univariate signals or images. However, given the nature of the data under consideration, adapting this concept to discrete surfaces is not straightforward, but is of great interest, especially when dealing with *compression*, since this type of techniques are more flexible than "single-rate" (i.e. non multiresolution-based) techniques. In this chapter, we will start by detailing the relationship between subdivision and multiresolution representations of discrete surfaces in Section 6.1. We will then review the existing single-rate compression techniques in Section 6.2. Progressive rate techniques, i.e. those based on a multiresolution representation of triangular meshes will be discussed in Section 6.3. A more detailed summary of such methods can be found in [63, part IV] and in [118, chap. 3]. A particular progressive rate hierarchical compression technique based on such representations will be described in Section 6.4. Modifications and improvements to this coding method will be detailed. Finally, the results using this method will be shown in Section 6.5.

6.1 Multiresolution representations of meshes

The definition of multiresolution for meshes relies on Definition 7, more precisely on the *nested space sequence* given in equation (2.14). If we recall the notation of Chapter 3 for discrete surfaces, i.e. a surface $\mathcal{M} = (V, E, F)$ is represented by a set of vertices, a set of edges and a set of faces. Under the assumption of dealing with a triangular mesh, it is sufficient to represent the mesh using a set of vertices V and a set of triangles T, i.e. $\mathcal{M} = (V, T)$. Following the idea of nested spaces, we can define the approximation $\mathcal{M}^j$ of $\mathcal{M}$ at level j, verifying

$$\mathcal{M}^0 \subset \mathcal{M}^1 \subset \cdots \subset \mathcal{M}^j \subset \cdots ,$$

where $\mathcal{M}^0$ is the coarsest approximation of $\mathcal{M}$. Following the multiresolution analysis formalism, the approximation at level $j+1$ is obtained using the relation

$$\mathcal{M}^{j+1} = \mathcal{M}^j \oplus \mathcal{D}^j,$$

where $\mathcal{D}^j$ represents the "details" needed to generate level $j+1$. The above relation really makes sense for "simple" data, such as for instance univariate signals or images, in which case $\mathcal{D}^j$ is the unique orthogonal complement to $\mathcal{M}^j$, which can be computed using filtering operations (i.e. scalar products). In the case of triangular meshes, this is not that obvious although it is formally possible to define such constructions. Pioneer work in this domain has been done by Lounsbery in [91], which already suggested to use a subdivision operator $\mathcal{S}$ to generate $\mathcal{M}^{j+1}$ from $\mathcal{M}^j$, i.e.

$$\mathcal{M}^{j+1} = \mathcal{S}\mathcal{M}^j.$$

Unlike the cases of univariate signals or images, it may be difficult, if not impossible, to build the details using a simple scalar product, especially when dealing with irregular meshes where the wavelet function is difficult to approximate simply. The interest of such representations for compression relies on assumptions similar to the ones made in the case of univariate signals and images, i.e. that the amplitude of the detail coefficients decreases exponentially when the level j increases. The most representative techniques used to build such hierarchical representations aimed at compressing 3D models will be detailed in Section 6.3.

6.2 Single-rate 3D model compression

In this section, we present the motivations and generalities with respect to compression of discrete surfaces. In this section (and in the whole chapter), we will limit ourselves to triangular meshes. Then we review the main single-rate compression techniques.

6.2.1 Compression generalities - Entropy coding

Let us first remind a classical result of information theory. Let us consider a signal, formed by *symbols* belonging to an *alphabet* $\mathcal{A} = \{\sigma_1, \dots, \sigma_n\}$ of size n. Under the assumption that the probability of appearance of each symbol σ_i, denoted by p_i is independent from the others, the average number of bits needed to represent the signal (using an optimal encoder) is given by the *entropy*,

$$-\sum_{i=1}^{n} p_i \log_2 p_i \text{ bits per symbol.}$$

While the underlying hypotheses are restrictive, this quantity is a good starting point for estimating the efficiency of a representation of a given signal. We refer the reader to [16] for more advanced concepts regarding information theory.

After choosing an appropriate representation of the signal, the resulting symbols are fed to a *coder* that turns them into a binary stream (called *bitstream*). One of the most popular coders is the *Huffman coding*, which associates a fixed-length binary codeword to a symbol. Compression efficiency is achieved by using as short as possible codewords for frequently appearing symbols (which requires to have an *a priori* knowledge of the statistics of the symbols). This type of coding has been used in image and video compression standards such as JPEG and MPEG. The other group of coders is formed by *arithmetic coders*, which uses a recursive interval subdivision in sub-intervals of length proportional to the probability of each symbol. The advantage of arithmetic coders is that they are efficient even for signals having an entropy smaller than 1 bit per symbol, however at the expense of a higher complexity than Huffman coding. The additional complexity of arithmetic coding is however not a true limitation, since many arithmetic coders have been used (under different flavors though), for instance in widely used compression standards such as JBIG[*] [65], JPEG[†] [64], JPEG 2000 [69], and MPEG 4 [67]. We refer the reader to [127] for an in-depth description and analysis of these coders.

Despite the simplicity of the representation of discrete surfaces using triangular meshes, the amount of information needed to represent a very detailed model can be huge. Almost all single-rate techniques consider separately the *geometry*, i.e. the coordinates of the vertices, and the *connectivity*, i.e. the full description of the edges connecting vertices together.

Connectivity can be represented by a list of vertex indices. Let us consider a mesh having F triangular faces and V vertices. In this case, $3F\lceil\log_2 V\rceil$ bits are needed to encode connectivity. Recalling the Euler formula from equation (3.11) for a graph, and assuming that the genus (i.e. the number of holes or handles in the graph) remains small (which leads to $F \approx 2V$), this quantity can be approximated by $6V\lceil\log_2 V\rceil$, which leads to a rate of $6\lceil\log_2 V\rceil$ bits per vertex. While this "upper-bound" is quite trivial to derive, a lower bound for connectivity coding has been proposed by Tutte in [130]. His study relied on the assumption that the graph representing the connectivity is planar (which is the case for meshes of genus zero), and shows that encoding a triangular graph requires at least $\log_2 \frac{256}{27} \approx 3.245$ bits per vertex. The first method to encode graphs with a constant number of bits per vertex was proposed by Turán in [129], which achieved a rate of 12 bits per vertex.

Regarding the geometry, a vertex is represented by 3 floating-point numbers for the x, y and z coordinates. Using the 32-bit representation of a floating point number, with a mantissa of 24 bits and an 8 bit exponent, gives usually far more precision than the one actually needed to represent the data. This amount of data can be reduced by quantizing appropriately the coordinates and using a prediction mechanism to exploit the spatial correlation between coordinates of neighboring vertices.

6.2.2 Main surface compression techniques

Let us now summarize the main single-rate compression techniques. We will not address mesh simplification techniques, such as the quadric error metric-based Qslim [53] already mentioned in the previous chapter.

Early work in this domain has been done by Deering [27]. His method was based on *generalized triangle strips*, compressed with a Huffman coder, to represent connectivity, and was aimed at transmitting faster the model to the graphics hardware throughout a bus having a limited bandwidth. Given the constraints on the hardware decoder, his method is very simple to decode. The geometry information is uniformly quantized. Typical values of the compression factor (with respect to the graphics primitives used to render the strip) are about 8 or 9 to 1, and rates between 25 to 30 bits per triangle are achieved. This method is now part of the Java3D API [122].

The first major step over Deering's method was proposed by Taubin and Rossignac in [125]. In their approach, referred to as *topological surgery*, the connectivity is represented by two trees, a *vertex tree* and a *face tree*. The vertex tree is the spanning tree of the graph connecting the vertices of the model together. The idea is to split each edge of this vertex spanning tree, to "flatten" the model into a simple polygon. The face tree is the spanning tree of the dual graph of this polygon, i.e. of the graph connecting the center of the triangles of the polygon together. Those trees, provided the

---

[*] QM-coder for version 1 and MQ-coder for version 2.

[†] The "common" JPEG coding makes use of Huffman coding, but a QM-arithmetic coder is used for lossless compression.

model is of genus zero, are sufficient to encode the connectivity information, and can be efficiently represented. Geometry is uniformly quantized and coded in the order of traversal of the vertex tree, using a linear prediction with previously encoded vertices. The connectivity and geometry are encoded using arithmetic coding. An alternate graph encoding technique for topological surgery has been suggested in [8]. Extensions have been proposed for handling holes and handles, and also a way to deal with non-manifold models in [57]. Typical rates needed to compress connectivity range between 1 to 8 bits per triangle. Adding geometry information (quantized over 10 to 12 bits per coordinate) yields a total rate of 6 to 11 bits per triangle. The improvement over Deering's method is significant, although this efficiency can drop for small models, because of spanning trees containing more "branching" nodes, which are more expensive to cost. It is interesting to note that the interleaving of connectivity and geometry information in the bitstream make a form of spatial progressive decoding possible. This method was initially proposed as a binary format for VRML and has been included almost "as-is" in the MPEG 4 standard [68] for compressing polygonal meshes.

An alternative compression method was proposed by Touma and Gotsman (TG) in [128]. Their idea is to exploit the fact that, in general, the 1-ring of a vertex is topologically equivalent to a disk. The algorithm encodes connectivity by using a restricted set of basic operations (`add`, `split` and `merge`). Each time a vertex is added, its valence is also encoded, hence the name of *vertex* or *valence-based coding* for this kind of algorithms. The geometry is coded using a *parallelogram predictor*, and the remaining error is quantized. Connectivity and geometry are encoded with a Huffman coder, using a run-length coding for connectivity and fixed-length coding for geometry. The results achieved with this coder are significantly better than those obtained with topological surgery. The compression rates of connectivity are typically of 0.5 to 1 bit per triangle and can drop significantly (0.1 bit per triangle) in the case of highly regular meshes, which is in any case lower than Tutte theoretical bound. However, in some pathological cases, coding the connectivity can lead to rates being worse than Tutte's bound, especially in the case of irregular meshes. Geometry coding requires 8 to 12 bits per vertex (using a quantization of coordinates over 8 bits). The overall gain of the compression rate with respect to topological surgery is in average 1.5. This method has been analyzed and improved by Alliez and Desbrun in [2], in which they also prove the upper bound of the entropy of their valence-based coding scheme is identical to Tutte's bound, i.e. 3.245 bits per vertex.

In [115], Rossignac proposed a face-based connectivity coding scheme, called *Edgebreaker*, which guarantees a worst case of 2 bits per triangle (for meshes of genus zero having no more than one boundary, although the method can handle meshes of higher genus and having more than one boundaries). As for valence-based coding, the connectivity is represented by a restricted set of operations (denoted by C, L, R, E and S letters), which are then encoded using a Huffman coder. The main difference with TG-like coding techniques is that each operation involves a face instead of a single vertex. The performance of Edgebreaker is of interest only in cases of irregular meshes, when valence-based coding reaches its worst cases. An improvement for regular meshes has been proposed in [123], which achieves 0.811 bits per triangle for regular (and large enough) meshes, which is still worse than TG. Moreover, the decoding step of Edgebreaker has a non-negligible complexity (in the worst case $O(n^2)$), although a decoding algorithm being only $O(n)$ has been proposed in [62].

6.3 Progressive mesh compression

Although the compression techniques reviewed in Section 6.2 are efficient and provide interesting rate-distortion performance, they often lack a certain flexibility. When dealing with highly detailed models, even a compressed version may require a non-negligible amount of bandwidth/CPU power to be transmitted/decoded, hence the idea of having a *multiresolution representation* of a surface. This approach has been successfully used for images (e.g. in the JPEG 2000 standard), and despite a compression efficiency that may not meet the one of single-rate techniques, the flexibility brought by multiresolution techniques (especially in terms of partial transmission/decoding) is worth studying. In this section we review the principal multiresolution techniques available for discrete surfaces, and associated compression methods.

Building a sequence of approximation of a triangular mesh is not a trivial operation, given the complexity of the data itself. However, a number of interesting approaches have been proposed regarding this topic, which are detailed here.

One of the earliest proposition in this direction was done by Hoppe in [60], and called *progressive meshes*. In this approach, the initial model was gradually simplified using *edge collapse* operations. All operations performed were stored, in order to be able to reconstruct the original mesh from the coarse one using the inverse operation (i.e. *vertex split*). Representing the connectivity using this technique requires $\lceil \log_2 V \rceil + 5$ bits per vertex (where V denotes the number of vertices), which is significantly higher than single rate techniques previously studied. This method was improved by Hoppe in [61]. This approach provides a very fine-grained level of details hierarchy, each vertex split is encoded separately. However, this is a drawback in terms of compression efficiency. A significant improvement of this method was proposed by Pajarola and Rossignac in [100]. Their idea was to encode batches of vertex split operations instead of encoding them one by one. This approach is conceptually closer to the multiresolution representation proposed in the previous section. The geometry was encoded using a vertex position prediction rule inspired from the Butterfly scheme and appropriate quantization of the remaining error. The cost of connectivity coding using this method dropped to about 3.5 bits per triangle, while the typical rate for geometry information was about 7.7 bits per triangle.

Another approach, referred to as *progressive forest split*, and inspired by progressive meshes was proposed by Taubin et al. in [124]. It follows some ideas proposed for topological surgery described in Section 6.2. The main idea is to use a *forest split*, i.e. a a group of vertex split operations to refine the mesh. The “forest” is simply a part of the vertex graph, and the forest split operation consists in splitting the edges of this forest, re-triangulate the resulting holes with simple polygons, and update the position of the new vertices. The forest is encoded using 1 bit per edge, and the polygon is encoded either using topological surgery or a fixed length coding using 2 bits per triangle. The total cost of connectivity encoding is about 4 bits per triangle, which is better than the initial proposal of progressive meshes. This method requires 1.7 to 2.7 times more bits for connectivity information than topological surgery, which is quite reasonable. It has been included in the MPEG 4 standard [68]. The variant of progressive meshes proposed by Pajarola and Rossignac is in general more efficient than progressive forest split, and produces less geometric distortions.

A different method, called *progressive geometry compression* (PGC), was proposed in [72], closer to a “classical” multiresolution framework. The approach to build the hierarchy of models is quite different from the above methods. The first step of PGC is to *remesh* the initial model, such that it has a semi-regular connectivity (i.e. vertices of valence 6 everywhere except in isolated locations). To achieve this, a parameterization is used, i.e. a bijective map between the original model $\mathcal{M}$ and the coarsest version $\mathcal{M}^0$ is built, using the MAPS algorithm [84]. In some applications where the exact connectivity of the original model should be preserved, this remeshing could be a problem, since original connectivity cannot be recovered. However, in most applications, this original connectivity loss is not an issue. The advantage of this approach (and any other remeshing-based approach) is that the connectivity of the j-th level is directly inferred from the base mesh, and does not need to be encoded explicitly. The only information remaining to encode is an offset (i.e. a vector) at each vertex (which is given by the parameterization). In order to be efficiently encoded this information is transformed using a *wavelet transform.* This wavelet transform is built so that Loop's scheme is the low-pass *reconstruction* filter, i.e the operation needed to obtain level $j + 1$ from level j. The corresponding high-pass filter is obtained using a quadrature-mirror construction (cf. [92] for a detailed presentation of quadrature-mirror filters). The forward wavelet transform (to obtain level j from level $j + 1$) is computed by solving a sparse linear system. The vector-valued wavelet coefficients are represented using a local frame defined by the local tangent plane, which improves greatly the distribution of those coefficients, and quantized. Those coefficients are then encoded using *zerotree coding* adapted from the SPIHT image coding algorithm [117], and then compressed using an arithmetic coder. The encoding method in this case differs significantly from the other progressive techniques, since it processes the wavelet coefficients by taking each bitplane separately. The method outperforms all other progressive and single rate coders. This performance is due partly to the connectivity which is no longer needed to be explicitly specified and partly to the very efficient representation of the geometry. The remeshing step is however non-trivial and the encoding is quite complex, since a sparse linear system has to be solved at each level. The complexity of the encoding step is however not a real limitation since the decoding step is much simpler (most applications will have the model to be encoded only once). This idea was pushed further in [58] by trying to represent the detail coefficients no longer by vectors but by a single scalar value (representing the offset in the normal direction). The performance of this extension remains unclear, the authors themselves mentioning that several troubles persist in such an approach. A theoretical study of a univariate version of this method has been proposed in [21].

Recently, a progressive method showing performance being close to PGC and coding losslessly the connectivity was proposed by Alliez and Desbrun [1]. This method is based on vertex removal to generate coarse level, and uses a valence-based coding, except that care is taken to be able to re-generate the original connectivity at decoding, using deterministic rules, but without encoding explicitly connectivity. Geometry information is represented using a local frame similar to the one used in PGC, and quantized. The results achieved are quite interesting since in average 1.8 bits per triangle are needed for connectivity (in case of regular meshes, this amount drops significantly). The overall size of the compressed model is only 10% above the one achieved using TG, with the benefit of being progressive.

Finally, a hierarchical method using subdivision as prediction mechanism has been proposed by Morán in [97]. We will detail this method in the next section.

6.4 Hierarchical 3D model coding with subdivision surfaces

In this section we present the approach followed in [97]. We also introduce modifications and extensions of this method to make use of the spherical subdivision scheme proposed in Chapter 4 as well as an alternate encoding method.

6.4.1 Method description

The main idea of this method is to start from a coarse version of the model, and generate the next level using surface subdivision as a prediction mechanism. At each newly inserted vertex, a detail vector is associated. As in PGC, the detail vector is represented in a local coordinate system (called *Frenet's coordinate frame*) defined by a local normal to the surface and two tangent vectors. This local frame is computed implicitly from the local geometry of the coarse model. The detail vector, instead of being computed using a parameterization, is estimated by minimizing a Hausdorff-like distance measurement to the original model (which in some cases can be troublesome). The detail vectors are quantized uniformly over their range (which has to be transmitted in the header of the compressed model). Instead of coding separately the three components of the detail vector, a single codeword is generated from the three components by interleaving the bits of the normal an tangential component. A fixed number of bits is given to represent each codeword, thus the amount is split between tangential and normal components, and privileges the normal component, but takes also into account the relative ranges of normal and tangential components. As shown in [72] for PGC, the normal component of the detail vector is the one that contains most energy, which justifies the bit-allocation step. Let us illustrate this bit-interleaving step by the example of a detail coded using 32 bits. 14 bits are dedicated for the normal component and 9 bits are used for each tangential components. Let us denote by n the quantized value of the normal component, by n_s its sign bit and by n_i its i-th bit. Similarly, we denote by t^1 and t^2 the quantized values of tangential details. We adopt the same notations to refer to the sign and i-th bits of these values. The resulting codeword is then

$$n_s t_s^1 t_s^2 \; n_{13} n_{12} n_{11} n_{10} n_9 n_8 t_8^1 t_8^2 n_7 t_7^1 t_7^2 \ldots n_1 t_1^1 t_1^2.$$

The resulting sequence is encoded using the SPIHT algorithm proposed by Said and Pearlman in [117]. In order to use SPIHT, the details codewords have to be organized into hierarchical trees. The detail hierarchy is different from the one used in PGC, since it is not *edge-based* but *face-based*. The resulting bitstream is then arithmetically encoded using the adaptive algorithm from Witten et al. [139].

Finally, as in PGC, the initial coarse mesh needs to be encoded separately. This represents only a minor offset in terms of compression rate, since the coarsest mesh can be compressed efficiently using of the method described in Section 6.2. In our tests, we will use the TG compression algorithm.

6.4.2 Proposed extensions

While the method described above is quite efficient and shows results being very close to the ones obtained with PGC (still without having to perform the relatively complex forward wavelet transform at encoding time), we propose several adaptations of the method.

Regarding the interpolating subdivision scheme used as prediction, we propose to replace the Butterfly scheme used by Morán by the spherical subdivision scheme proposed in Chapter 4.

The SPIHT-based entropy coding, while being quite efficient in terms of compression rate, lacks in terms of capabilities for progressive decoding. Since the hierarchy in SPIHT is organized using the layout of detail trees, decoding a single level of resolution is not a trivial task. In order to overcome this restriction, we propose to adapt the encoding method used in JPEG 2000 to our case. We refer the reader to [108, 126, 127] for a description of this algorithm. This method, while being slightly more complex and in average less efficient than SPIHT, provides the flexibility we need at decoding time, i.e. providing both partial decoding of bitplanes and partial decoding of levels of resolution. It also provides a sound basis for adding error-resilience capabilities to the considered method (e.g. by resetting the arithmetic coder statistics at each level and inserting synchronization markers). This algorithm has been adapted to suit the type of data considered.

Let us now describe this method in details. In order to adapt this method, the original arithmetic coder used must be replaced by a context-based arithmetic coder. We chose the same coder used in JPEG 2000, i.e. the MQ-coder [127]. The encoding method, also bitplane-based, consists in three passes: significance propagation, refinement and cleanup. Each pass encodes a part of the bitplane. Two boolean flags per codeword are needed, one indicating whether it is already significant (initialized to FALSE), i.e. if a bit in higher bitplanes is equal to one, and one indicating if this is the first refinement (initialized to TRUE). The coding passes are repeated for each level of resolution and for each magnitude bitplane. At the end of each bitplane, the MQ-coder statistics are reset and a marker is put in the bitstream, so that partial decoding of the bitplanes can be easily achieved by searching the markers. Each pass loops over all vertices/triangles belonging to the same level of resolution (using a list created and updated when subdividing triangles) and encodes the associated detail codeword.

1. *Significance propagation pass.*
   If the current codeword is not significant and has at least one significant neighbor, the current bit is encoded according to the number of significant neighbors belonging to the same level. In our implementation, we have used 4 contexts: 3 for the cases where the current vertex has 1, 2 or 3 significant neighbors and the last one for the case where the current vertex has more than 3 significant neighbors. If this bit is equal to 1, the codeword is marked as significant and the sign bits of the detail codeword are encoded. One context is used for the sign bit of the normal detail, and three contexts for the sign of tangential components (since the 2 bits they are coded simultaneously, the MQ coder needs $2^2 - 1 = 3$ contexts).

2. *Refinement pass.*
   If the current bit has been processed during the significance propagation pass it is skipped. If the current codeword is not significant, it is skipped and will be processed in the cleanup pass. On the contrary, if the codeword is not refined for the first time, the current bit is encoded using a specific context. If the codeword is refined for the first time, we set a flag to indicate it has now been refined and encode the bit using two contexts. In this case the context chosen depends on the fact that this codeword has at least one significant neighbor or not.

3. *Cleanup pass.*
   Finally all remaining bits are processed in this cleanup pass. The current bit is encoded using a special context, and if a non-significant yet codeword has a bit equal to 1, the sign bits are encoded, and the codeword marked as significant.

While needing to encode explicitly all bits of each codeword (whereas SPIHT implicitly represent all no-longer significant detail trees), this simplified version of the JPEG 2000 coder allies high compression rate and easy partial decoding (both of levels and bitplanes).

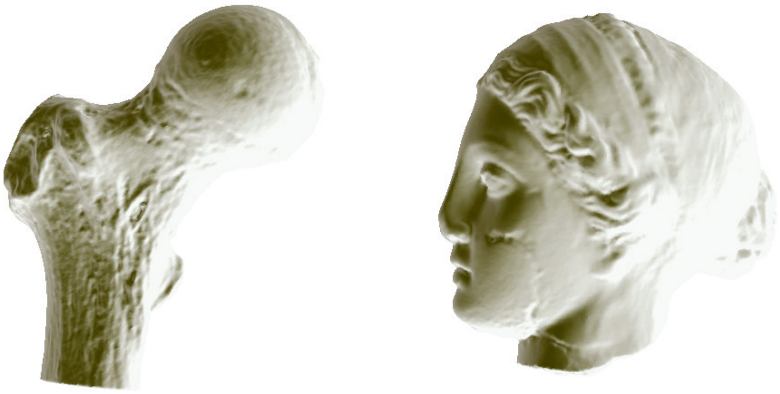
6.5 Results

In this section we detail various results achieved using the proposed compression method. In order to evaluate the performance of the compression method and of the extensions, we will study the behavior of its rate-distortion in several conditions. As in Chapter 5, the distortion between the original and compressed model will be estimated using the Hausdorff-based metric [3]. A first series of tests will aim at determining the influence of subdivision scheme on compression efficiency, and will be detailed in Section 6.5.1. Finally a performance comparison between the original SPIHT-based method and the JPEG 2000-like coder will be made in Section 6.5.2.

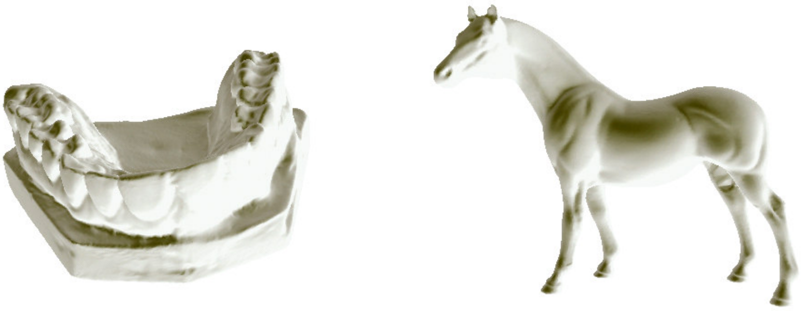
The available datasets for these tests is quite restricted, given the nature of the detail computation method. In fact, since at each newly inserted vertex, the associated detail is estimated using a Hausdorff-like distance to the original surface, cases where high curvature details are not recovered occur. This erratic behavior is highly dependent on the shape of the original model, and the quality of initial coarse meshes (generated using Qslim [53]). This could be solved by using a parameterization method, e.g. [29, 71], to compute more accurately the detail coefficients. In our experiments, we use four highly detailed models

- the *Balljoint* model (already used in Chapter 5), which has 137062 vertices and 274120 triangles. The coarsest initial mesh has 102 vertices and 200 triangles. Given the exponential growth of the number of triangles, we can use 5 levels of subdivision.
- the *Venus* model, which is a high-resolution scan of a statue. It has 134345 vertices and 268686 triangles. The initial coarse mesh has 52 vertices and 100 triangles. This model can be subdivided 5 times.
- the *Teeth* model, which has 116604 vertices and 233204 triangles. The initial coarse mesh has 52 vertices and 100 vertices, which also bounds the maximum subdivision level to 5.
- the *Horse* model, which has 48485 vertices and 96966 triangles. The initial coarse mesh has 165 vertices and 326 triangles. This model can be encoded using 4 levels of subdivision.

The aspect of these models is shown in Figure 6.1.



(a) Balljoint model **(b)** Venus model



(c) Teeth model **(d)** Horse model

Figure 6.1: Original high-resolution test models

In order to evaluate the influence of the subdivision scheme and of the proposed coder, the models are encoded using (at most) 32 bitplanes. Rate-distortion curves are obtained by decoding only a smaller number of bitplanes. The size of the base model compressed using the TG coder (using 12 bits for the geometry) is added to the size of the compressed details. The informations regarding the compressed base models (size and distortion) are summarized in Table 6.1. When available, the results achieved using the proposed method will be compared to PGC, i.e. for the Venus and Horse model.

Model	Compressed size (bytes)	Relative ℓ_2-error (% of the bounding box diagonal length)
Balljoint	559	0.7
Venus	371	1.23
Teeth	377	1.23
Horse	868	1.29

Table 6.1: Compressed initial coarse meshes - size and distortion

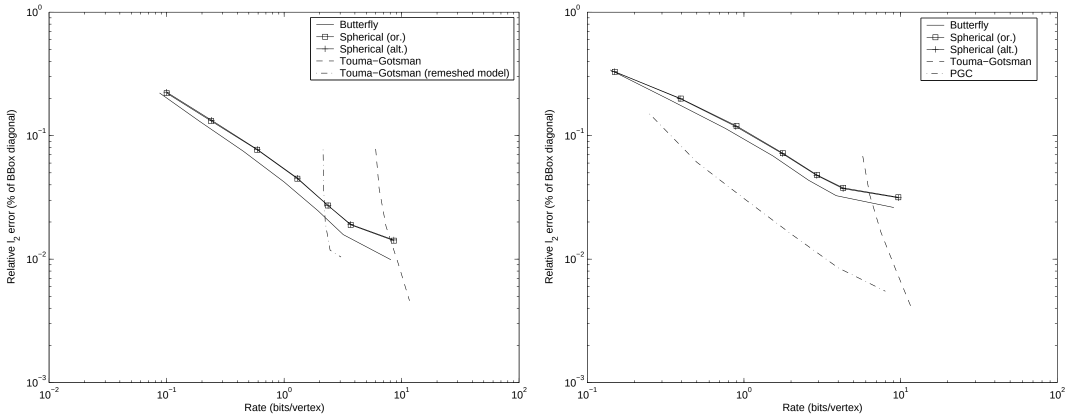
6.5.1 Influence of the subdivision scheme

The rate-distortion curves obtained using the proposed method, for the different subdivision schemes considered are shown in Figure 6.2. As a comparison, the rate-distortion for the original model encoded with the TG coder using 8 to 12 bits for the geometry, is also shown.

The rate-distortion curves for the considered method show that the considered compression method gives very similar results, both with the Butterfly and spherical subdivision scheme. In general, the Butterfly scheme has a slight advantage against the spherical scheme (although the contrary happens for the Teeth model). As suggested in Chapter 5, the difference between the two flavors of spherical subdivision scheme are negligible.

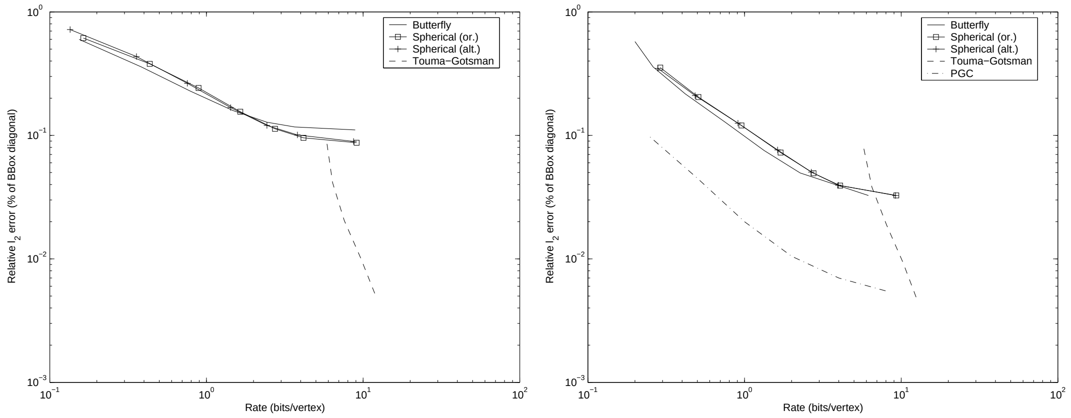
The performance with respect to the TG coder is of course more interesting at low bitrates. At high rates, the TG coder is usually better than the proposed technique, but given the small possible range of bits allocated for geometry information, low bitrates are not reachable without simplifying the model, or at least remeshing it to take advantage of the performance of TG on a mesh with regular connectivity, prior to encoding. The greater flexibility of the proposed coding method in terms of progressive decoding is clearly an advantage over single-rate techniques. As shown for the Balljoint model, a pre-remeshing improves the compression efficiency of the TG coder but the performance of the proposed progressive technique at low bitrate seem out of reach without simplifying the original model.

When available (i.e. for the Horse and Venus models) the comparison with PGC shows clearly that the proposed technique is less efficient in terms of rate-distortion, at the expense of a much more complex encoding algorithm for PGC. It is also likely that the proposed method misses a parameterization in order to compute the exact detail vectors and ensure properly the convergence to the original surface. During the tests, we have observed that when using a coarse model not being close enough to the original surface, the Hausdorff-based detail computation may fail completely, especially where "spikes" have been removed during the simplification process, and that the quality of the original coarse model affects the reconstruction error and the coding efficiency. It is likely that using a parameterization step, the residual distortion at high bitrates should be lower than the ones we achieved in our tests. As stated in [72], the rate-distortion curve of the TG coder intersects the one of PGC at a distortion value that roughly corresponds to the remeshing error. In our



(a) Balljoint

(b) Venus



(c) Teeth

(d) Horse

Figure 6.2: Rate-distortion performance of the proposed coder with respect to subdivision scheme

experiments, the remeshing error introduced by the proposed method are significantly higher than the ones achieved with PGC. It is therefore likely that this remeshing error perturbates the rate-distortion curves of the proposed scheme (e.g. the Teeth model). This supposition is supported by the maximum value of the distortion between the decompressed model and the original one, which is significantly higher when using spherical schemes. As a consequence, local distortions introduced by the detail computation method disturb the convergence of the method toward the original surface. Therefore the experiments do not permit to conclude safely on whether changing the subdivision scheme in our method is really an advantage or not, since the Butterfly and spherical rate-distortion curves are very close to each other.

6.5.2 SPIHT vs. JPEG 2000-like detail encoding

In this section we compare the performance of the JPEG 2000-like encoding method to the original SPIHT-based approach. Although the JPEG 2000-like method is more complex, this overhead is negligible when compared to the details computation time. The original SPIHT approach has also been modified to use the MQ arithmetic coder used in the JPEG 2000-like coder. The results giving the different details file (using the Butterfly scheme) sizes for the considered models, are presented in Table 6.2.

	SPIHT+Witten coder	SPIHT+MQ coder	JPEG 2000-like coder	JPEG 2000-like coder without markers
Balljoint	103645	101442	103188	102868
Venus	58679	55732	57657	57337
Teeth	57286	54030	56658	56338
Horse	32929	31057	31409	31153

(a) 32 bitplanes

	SPIHT+Witten coder	SPIHT+MQ coder	JPEG 2000-like coder	JPEG 2000-like coder without markers
Balljoint	12402	12402	11954	11794
Venus	9174	9250	9205	9045
Teeth	8458	8431	8657	8497
Horse	2802	2836	2845	2717

(b) 16 bitplanes

	SPIHT+Witten coder	SPIHT+MQ coder	JPEG 2000-like coder	JPEG 2000-like coder without markers
Balljoint	550	552	565	485
Venus	485	489	529	449
Teeth	574	575	611	531
Horse	124	126	178	114

(c) 8 bitplanes

Table 6.2: Details file size (in bytes) for the different coders used, using 8, 16 and 32 bitplanes as codeword length.

The comparison between the different coding methods shows an advantage for SPIHT, which is not surprising given that unsignificant detail trees are not coded at all, whereas the JPEG 2000-like coder needs to code explicitly every bit. At high bitrates (32 bitplanes), the overhead of the JPEG 2000-like coder with respect to SPIHT is really negligible and varies roughly between 1 and 5% of the bitstream size (among which the markers also play a non-negligible part). In this case, the MQ coder is also more efficient than the one originally used.

When going to lower bitrates (16 and 8 bitplanes), SPIHT has a clear advantage over the JPEG 2000-like coder. The MQ coder is also in general less efficient than the original one, as the smaller number of bits to encode may not allow the statistics to converge fully. However, the disadvantage of the JPEG 2000-like coder is mostly because of the two-bytes markers inserted at the end of each bitplane. For a stream coding 8 bitplanes on four levels of subdivision (as in the Horse model), this represent 64 bytes, which is far from negligible and may represent up to one third of the total stream size. As shown by the last column in Table 6.2 which removes the size of the markers used in the JPEG 2000-like coder, this coder is really more efficient than SPIHT. The markers are needed to achieve partial decoding. However, if choosing at encoding time, as in JPEG 2000, if the bitstream should be organized in a SNR-progressive or level-progressive order, some of these markers can be omitted. Another solution is to embed the length of information coded for each bitplane into the header of the bitstream, as it is also done in JPEG 2000.

6.6 Conclusion

The proposed method provides an efficient way of using interpolatory surface subdivision schemes in order to achieve discrete surface progressive compression. Given the implicit representation of mesh connectivity, the method provides better rate-distortion performance than single-rate techniques. The proposed method clearly lacks a parameterization step in order to ensure convergence to the original model. Given this remeshing error which remains higher than the one achieved with PGC, it is not possible to conclude whether replacing Butterfly scheme by the spherical one is really worth, as the difference seen in terms of rate-distortion could be caused by the remeshing error introduced by the Hausdorff-based detail computation method. The Butterfly scheme seems however to have an advantage over the proposed non-linear technique. The JPEG 2000-based encoder has proved to be very efficient and scalable, since in terms of pure compression efficiency (if the markers are removed) and flexibility, it provides better results than SPIHT. The double scalability at decoding, both SNR and level progressive, makes the overhead added in terms of bitrate quite acceptable.

Finally, it is worth remarking that using a surface subdivision method to predict new vertex position is not optimal at all when compressing model having lots of sharp edges (e.g. mechanical pieces), since the subdivision scheme generates smooth limit surfaces. In this case, tagging *crease edges*, and using an adapted version of the subdivision scheme around those edges (cf. for instance [6]) would be useful to improve compression efficiency.

Conclusions

7

7.1 Achievements

In this thesis, an original non-linear interpolatory subdivision scheme has been proposed, suitable for both regular and irregular initial grids. Given the local coordinate system used to compute the newly inserted points, this scheme can be trivially adapted to discrete triangulated surfaces, and also supports meshes with boundaries. Although being more complex than a linear scheme of identical support, the proposed method remains quite reasonable in terms of computation time.

It has been proved, using a direct proof, that in the univariate case, the scheme converges uniformly toward continuous functions. However, the techniques used to study linear subdivision schemes are not directly applicable to the proposed scheme, and the ongoing works regarding non-linear subdivision are also focused on relatively restricted classes of non-linear schemes. In order to be able to study the proposed scheme, we have proposed several numerical criteria to check the convergence, regularity of the limit functions and approximation order. Those estimates provide strong evidences that the proposed scheme behaves very closely to the four-point scheme.

When adapted to subdivide discrete surfaces, the proposed scheme also behaves closely to the Butterfly scheme (which generalizes the univariate four-point scheme to triangular meshes). The remarks made regarding the difficulty of studying analytically the proposed univariate scheme still hold for its adaptation to discrete surfaces. Therefore, a subset of numerical criteria used in the univariate case is adaptable to the surface case, and also confirms this behavior. In addition, the study of the Gaussian and mean curvatures confirms that the proposed non-linear scheme gives results being closer to the theoretical values in the case of the torus and icosahedron than the Butterfly scheme.

The proposed scheme has been included in an existing multiresolution compression method and used as a predictor. Here again, its performance in terms of rate-distortion are very close to the ones achieved using an interpolatory subdivision scheme having an identical support, namely Butterfly scheme. In addition to this, a detail coding method for this compression techniques, has been adapted from the one used in the JPEG 2000 image compression standard. The proposed coder shows performance comparable to the original coding method based on SPIHT, and provides a convenient multiresolution representation in both SNR-progressive and level-progressive way, which is an improvement over SPIHT's SNR-only progressivity.

7.2 Future works

As stated before, theoretical works concerning non-linear subdivision schemes are far from being as complete as the studies concerning the linear case. The proposed scheme does not fit into the classes of non-linear schemes studied in the literature. While the proposed numerical criteria provide interesting evidences and are easy to apply, they do not replace an analytic proof. They however help to formulate a number of conjectures regarding the proposed univariate scheme, i.e. that it converges to at least $\mathcal{C}^1$ and possibly to $\mathcal{C}^{2-\varepsilon}$ functions. Its approximation order should also be close to the one of the four-point scheme, i.e. 4. The surface subdivision method seem also to lead to $\mathcal{C}^1$ limit surfaces. Such conjectures still to need to be proved (or proved wrong).

In the proposed subdivision scheme, the function h has been designed in a relatively intuitive way, which does not claim to be optimal. There are probably other choices possible for this function that may be suitable, or even give better results. The optimal design for h is of course linked to a complete theoretical analysis of the proposed scheme.

The numerical criteria proposed for the univariate schemes give a relatively complete overview of the behavior of a scheme. However, given their need for a knowledge of a "sampling-step", the regularity of the limit surface and approximation order of a surface subdivision scheme are not trivial to adapt from the univariate case. The development of such criteria for surface subdivision scheme would certainly be useful to compare further those methods.

We have proposed an interpolatory binary non-linear scheme. However, it may be possible to adapt the proposed idea, i.e. to use a local coordinate system, to derive n-ary schemes and/or approximating schemes. In the case of surfaces, we have seen that $\sqrt{3}$-schemes have the advantage of increasing the number of triangles at each subdivision step by a factor 3 instead of 4 for the usual binary case. Maybe such a scheme could be designed using a non-linear method similar to the one we proposed.

As we have seen, surface subdivision is a handy tool to create multiresolution representations of discrete surfaces. This representation is of great interest when compressing highly detailed models. The compression scheme we studied here has proved to be quite efficient. However, given the detail computation method used, the final remeshing error is significantly higher than in similar method that make use of a parameterization. As a consequence the rate-distortion performance of the compression method we used are lower than what could be theoretically achieved using a parameterization. Moreover, this remeshing error does not permit to conclude on whether using our non-linear scheme instead of the Butterfly scheme improves the rate-distortion or not, hence the need for this parameterization step. However, the results are already quite promising, and confirm that the encoding method is of interest, as it allies efficiency and scalability. Moreover, incorporating a $\sqrt{3}$-scheme into the studied compression method could lead to a finer control of the resolution of the reconstructed model, given the slightly lower growth of the number of triangles at each subdivision step.

Finally, although the ℓ_2 and ℓ_∞ errors for discrete surfaces based on the Hausdorff distance provide a reliable distortion measurement, they suffer from the same drawbacks than univariate or two-dimensional ℓ_2 and ℓ_∞ error metrics such as PSNR. In particular they do not provide indications on the visual quality, since for instance, small ripples on a surface may not affect significantly the ℓ_2 error, whereas they may become truly disturbing artifacts. An error metric taking into account such features would be a valuable tool for finer analysis.

Bibliography

[1] P. Alliez, M. Desbrun (2001). Progressive compression for lossless transmission of triangle meshes. In *Proceedings of SIGGRAPH'01*, pp. 195–202, ACM Press.

[2] P. Alliez, M. Desbrun (2001). Valence-driven connectivity encoding for 3D meshes. *Computer graphics forum* **20**(3):480–489. Presented at EUROGRAPHICS 2001, 4-7 Sept. 2001, Manchester, UK.

[3] N. Aspert, D. Santa-Cruz, T. Ebrahimi (2002). MESH: Measuring Errors between Surfaces using the Hausdorff distance. In *Proceedings of the IEEE International Conference on Multimedia and Expo*, vol. I, pp. 705 – 708. `http://mesh.epfl.ch`.

[4] A. A. Ball, D. J. T. Storry (1986). A matrix approach to the analysis of recursively generated B-spline surfaces. *Computer-Aided Design* **18**(8):437–442.

[5] A. A. Ball, D. J. T. Storry (1988). Conditions for tangent plane continuity over recursively generated b-spline surfaces. *ACM Transactions on Graphics* **7**(2):83–102.

[6] H. Biermann, A. Levin, D. Zorin (2000). Piecewise smooth subdivision surfaces with normal comtrol. In *Proc. of the International Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 2000*, pp. 113–120.

[7] W. Böhm, G. Farin, J. Kahmann (1984). A survey of curve and surface methods in CAGD. *Computer Aided Geometric Design* **1**:1–60.

[8] F. Bossen (1999). *On the art of compressing three-dimensional polygonal meshes and their associated properties*. Ph.d. dissertation, École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland. No. 2012.

[9] E. Catmull, J. Clark (1978). Recursively generated B-spline surfaces on arbitrary topological meshes. *Computer Aided Design* **10**(6):350–355.

[10] A. Cavaretta, W. Dahmen, C. Micchelli (1991). *Stationary Subdivision*, vol. 93 of *Memoirs of the AMS*. American Mathematical Society.

[11] G. M. Chaikin (1974). An algorithm for high-speed curve generation. *Computer Graphics and Image Processing* **3**:346–349.

[12] P. Cignoni, C. Rocchini, R. Scopigno (1998). Metro: measuring error on simplified surfaces. *Computer Graphics Forum* **17**(2):167–174.

[13] J. Claes, K. Beets, F. van Reeth (2002). A corner-cutting scheme for hexagonal subdivision surfaces. In *Proceedings of the Shape Modeling International 2002 (SMI'02)*, pp. 13–20, IEEE Computer Society.

[14] A. Cohen, N. Dyn (1996). Nonstationary subdivision schemes and multiresolution analysis. *SIAM Journal of Mathematical Analysis* **27**(6):1745–1769.

[15] A. Cohen, N. Dyn, B. Matei (2001). Quasilinear Subdivision Schemes with Applications to ENO Interpolation. Preprint - submitted to Applied and Computational Harmonic Analysis.

[16] T. M. Cover, J. A. Thomas (1991). *Elements of Information Theory.* Wiley.

[17] I. Daubechies, I. Guskov, W. Sweldens (1999). Regularity of irregular subdivision. *Constructive Approximation* **15**(3):381–426.

[18] I. Daubechies, I. Guskov, W. Sweldens (2001). Commutation for irregular subdivision. *Constr. Approx.* **17**(4):479–514.

[19] I. Daubechies, J. Lagarias (1991). Two-scale difference equations. I. Existence and global regularity of solutions. *SIAM Journal of Mathematical Analysis* **22**:1338–1410.

[20] I. Daubechies, J. Lagarias (1992). Two-scale difference equations. II. Local regularity, infinite product of matrices, and fractals. *SIAM Journal of Mathematical Analysis* **23**:1031–1079.

[21] I. Daubechies, O. Runborg, W. Sweldens (2002). *Normal multiresolution approximation of curves.* Tech. rep., Princeton University.

[22] C. de Boor (1987). Cutting corners always works. *Computer Aided Geometric Design* **4**:125–131.

[23] C. de Boor, K. Höllig, S. Riemenschneider (1993). *Box splines*, vol. 98 of *Applied mathematical sciences.* Springer-Verlag.

[24] G. de Rham (1947). Un peu de mathématiques à propos d'une courbe plane. *Revue de Mathématiques Élémentaires* **II**. Oeuvres complètes, pp. 678-689.

[25] G. de Rham (1956). Sur une courbe plane. *Journal de Math. Pures et Appliquées* **XXXV**(1):25–42.

[26] G. de Rham (1959). Sur les courbes limites de polygones obtenus par trisection. *Enseignement Mathématique* **V**(1). Oeuvres complètes, pp. 729-743.

[27] M. Deering (1995). Geometry compression. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pp. 13–20, ACM Press.

[28] G. Derfel, N. Dyn, D. Levin (1995). Generalized refinement equations and subdivision processes. *Journal of Approximation Theory* **80**(2):272–297.

[29] M. Desbrun, M. Meyer, P. Alliez (2002). Intrinsic parameterizations of surfaces meshes. *Computer Graphics forum* **21**(3). Eurographics 2002 Conference Proceedings.

[30] G. Deslauriers, S. Dubuc (1987). Interpolation dyadique. In G. Cherbit (ed.), *Fractals: Dimensions non entières et Applications*, pp. 44–55, Masson, Paris.

[31] G. Deslauriers, S. Dubuc (1989). Symmetric Iterative Interpolation Processes. *Constructive Approximation* **5**:49–68.

[32] D. Doo, M. Sabin (1978). Behavior of recursive division surfaces near extraordinary points. *Computer Aided Design* **10**(6):356–360.

[33] D. W. H. Doo (1978). A subdivision algorithm for smoothing down irregular shaped polyhedrons. In I. C. Soc. (ed.), *Proceedings of Intl. Conf. on Interactive Techniques in Caomputer Aided Design*, pp. 157–165.

[34] S. Dubuc (1986). Interpolation through an iterative scheme. *Journal of Mathemtical Analysis and Applications* **114**:185–204.

[35] N. Dyn (1992). Subdivision schemes in computer-aided geometric design. In W. Light (ed.), *Advances in Numerical Analysis*, vol. II, pp. 36–104, Clarendon Press.

[36] N. Dyn, J. Gregory, D. Levin (1987). A 4-point interpolatory subdivision scheme for curve design. *Computer Aided Geometric Design* **4**:257–268.

[37] N. Dyn, J. Gregory, D. Levin (1991). Analysis of linear binary subdivision schemes for curve design. *Constructive Approximation* **7**:121–147.

[38] N. Dyn, J. A. Gregory, D. Levin (1995). Piecewise uniform subdivision schemes. In M. Daehlen, T. Lyche, L. Schumaker (eds.), *Mathematical Methods for Curves and Surfaces*, pp. 111–119, Vanderbilt University Press.

[39] N. Dyn, S. Hed, D. Levin (1993). Subdivision schemes for surface interpolation. In A. C. et al. (ed.), *Workshop on Computational Geometry*, pp. 97–118.

[40] N. Dyn, F. Kuijt, D. Levin, R. van Damme (1999). Convexity preservation of the four-point interpolatory subdivision scheme. *Computer Aided Geometric Design* **16**(8):789–792.

[41] N. Dyn, D. Levin (1986). Smooth interpolation by bisection algorithms. In C. K. Chui, L. L. Schumaker, J. Ward (eds.), *Approximation Theory*, vol. V, pp. 335–337, Academic Press.

[42] N. Dyn, D. Levin (1992). Stationary and non-stationary binary subdivision schemes. In T. Lyche, L. L. Schumaker (eds.), *Mathematical Methods in Computer-Aided Geometric Design II*, pp. 209–216, Academic Press, Inc.

[43] N. Dyn, D. Levin (1994). The subdivision experience. In P. Laurent, A. L. Méhauté, L. Schumaker (eds.), *Wavelets, Images and Surface Fitting*, pp. 229–244, A. K. Peters, Wellesley, MA.

[44] N. Dyn, D. Levin (1995). Analysis of asymptotically equivalent binary subdivision schemes. *Journal of Mathematical Analysis and Applications* **193**:594–621.

[45] N. Dyn, D. Levin (2002). Subdivision schemes in geometric modelling. *Acta Numerica* **11**(00):73–144.

[46] N. Dyn, D. Levin, J. A. Gregory (1990). A butterfly subdivision scheme for surface interpolation with tension control. *ACM Transactions on Graphics* **9**(2):160–169.

[47] N. Dyn, D. Levin, D. Liu (1992). Interpolatory convexity-preserving subdivision schemes for curves and surfaces. *Computer-Aided Design* **24**(4):211–216.

[48] N. Dyn, D. Levin, D. Liu (1992). nterpolatory convexity-preserving subdivision schemes for curves and surfaces. *Computer-Aided Design* **24**(4):211–216.

[49] N. Dyn, D. Levin, C. A. Micchelli (1990). Using parameters to increase smoothness of curves and surfaces generated by subdivision. *Computer Aided Geometric Design* **7**:129–140.

[50] G. Farin (1996). *Curves and surfaces for computer aided geometric design.* Academic Press, 4th edn.

[51] G. E. Farin (1999). *NURBS: from projective geometry to practical use.* A. K. Peters, Ltd., Natick, Massachusetts, 2nd edn.

[52] M. S. Floater, C. A. Michelli (1998). Nonlinear Stationary Subdivision. In N. K. Govil et al. (eds.), *Approximation theory : in Memory of A. K. Varma*, pp. 209–224, Marcel Dekker, New-York.

[53] M. Garland, P. S. Heckbert (1997). Surface simplification using quadric error metrics. In *Proceedings of SIGGRAPH'97*, pp. 209–216. `http://graphics.cs.uiuc.edu/~garland/software/qslim20.html`.

[54] J. A. Gregory (1991). An introduction to bivariate uniform subdivision. In D. F. Griffiths, G. A. Watsin (eds.), *Numerical Analysis 1991*, Pitman Research Notes in Mathematics, pp. 103–117, Longman Scientific and Technical.

[55] J. A. Gregory, R. Qu (1996). Nonuniform corner cutting. *Computer Aided Geometric Design* **13**(8):763–772.

[56] X. Gu, S. Gortler, H. Hoppe (2002). Geometry images. In *Proceedings of SIGGRAPH'02*, pp. 355–361.

[57] A. Guéziec, G. Taubin, F. Lazarus, W. Horn (1998). Converting sets of polygons to manifold surfaces by cutting and stitching. In *IEEE Visualization 98 Conference Proceedings*, pp. 383–390 and 553.

[58] I. Guskov, K. Vidimče, W. Sweldens, P. Schröder (2000). Normal meshes. In *Proc. of the International Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 2000*, pp. 95–102, New Orleans, LA, USA.

[59] M. Hassan, I. Ivrissimtzis, N. Dodgson, M. Sabin (2002). An interpolating 4-point $\mathcal{C}^2$ ternary stationary subdivision scheme. *Computer Aided Geometric Design* **19**(1):1–18.

[60] H. Hoppe (1996). Progressive meshes. In *Proceedings of SIGGRAPH'96*, pp. 99–108, ACM.

[61] H. Hoppe (1998). Efficient implementation of progressive meshes. *Computer and Graphics* **22**(1):27–36.

[62] M. Isenburg, J. Snoeyink (2001). Spirale Reversi: reverse decoding of the Edgebreaker encoding. *Computational Geometry* **20**(1-2):39–52.

[63] A. Iske, E. Quak, M. S. Floater (eds.) (2002). *Tutorials on Multiresolution in Geometric Modelling.* Mathematics and Visualization. Springer-Verlag.

[64] ISO/IEC (1993). *ISO/IEC 10918:1994 Information technology — Digital compression and coding of continuous-tone still images: Requirements and guidelines.*

[65] ISO/IEC (1993). *ISO/IEC 11544:1993 Information technology — Coded representation of picture and audio information — Progressive bi-level image compression.*

[66] ISO/IEC (1998). *ISO/IEC 14772-1:1998 Information technology — Computer graphics and image processing — The Virtual Reality Modeling Language (VRML) — Part 1: Functional specification and UTF-8 encoding.*

[67] ISO/IEC (1999). *ISO/IEC 14496-2:1999: Information technology — Coding of audio-visual objects — Part 2: Visual.*

[68] ISO/IEC (2000). *ISO/IEC 14496-2:1999/Amd 1:2000: Visual extensions.*

[69] ISO/IEC JTC 1/SC 29/WG 1 (2000). *ISO/IEC FDIS 15444-1: Information technology — JPEG 2000 image coding system: Core coding system [WG 1 N 1890].*

[70] S. Karbacher, S. Seeger, G. Häusler (2000). A non-linear subdivision scheme for triangle meshes. In G. Greiner, H. Niemann, H.-P. Steinel, B. Girod (eds.), *Vision Modeling and Visualization*, pp. 163–170.

[71] A. Khodakovsky, N. Litke, P. Schröder (2003). Globally smooth parameterizations with low distortion. In *Proceedings of SIGGRAPH'03.* To appear.

[72] A. Khodakovsky, P. Schröder, W. Sweldens (2000). Progressive geometry compression. In *Proc. of the International Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 2000*, pp. 271–278, New Orleans, LA, USA.

[73] L. Kobbelt (1995). Interpolatory refinement as low pass filter. In M. Daehlen, T. Lyche, L. L. Schumaker (eds.), *Mathematical Methods for Curves and Surfaces*, pp. 281–290, Vanderbilt University Press.

[74] L. Kobbelt (1996). Interpolatory subdivision on open quadrilateral nets with arbitrary topology. In *Computer Graphics Forum (Proc. EUROGRAPHICS '96), 15(3)*, vol. 15, pp. 409–420. Eurographics '96 issue.

[75] L. Kobbelt (1996). A variational approach to subdivision. *Computer Aided Geometric Design* **13**:743–761.

[76] L. Kobbelt (2000). $\sqrt{3}$-Subdivision. In *Proceedings of SIGGRAPH*, pp. 103–112, ACM SIGGRAPH.

[77] E. Kreyszig (1991). *Differential geometry.* Dover publications, Inc. Reprint of the edition published by University of Toronto press, 1959.

[78] F. Kuijt (1998). *Convexity Preserving Interpolation – Stationary Nonlinear Subdivision and Splines.* Ph.D. thesis, University of Twente, Faculty of Mathematical Sciences.

[79] F. Kuijt, R. van Damme (1997). Smooth interpolation by a convexity preserving lon-linear subdivision algorithm. In A. L. Méhauté, C. Rabut, L. Schumaker (eds.), *Surface Fitting and Multiresolution Methods*, pp. 219–224, Vanderbilt University Press, Nashville, TN.

[80] F. Kuijt, R. van Damme (1998). Convexity preserving interpolatory subdivision schemes. *Constructive Approximation* **14**(4):609–630.

[81] F. Kuijt, R. van Damme (1999). Monotonicity preserving interpolatory subdivision schemes. *Journal of Computational and Applied Mathematics* **101**:203–229.

[82] F. Kuijt, R. van Damme (2002). Shape preserving interpolatory subdivision schemes for nonuniform data. *Journal of Approximation Theory* **114**(1):1–32.

[83] U. Labsik, G. Greiner (2000). Interpolatory $\sqrt{3}$-subdivision. *Computer Graphics Forum (Proc. of Eurographics 2000)* **19**(3).

[84] A. W. F. Lee et al. (1998). MAPS: Multiresolution Adaptive Parameterization of Surfaces. In *Proceedings of SIGGRAPH*, pp. 95–104, ACM SIGGRAPH.

[85] A. Levin (2003). Polynomial generation and quasi-inteprolation in stationary non-uniform subdivision. *Computer Aided Geometric Design* **20**(1):41–60.

[86] D. Levin (1999). Using Laurent polynomial representation for the analysis of non-uniform binary subdivision schemes. *Advances in Computational Mathematics* **11**(1):41–54.

[87] M. Levoy et al. (2000). The digital Michelangelo project: 3D scanning of large statues. In *Proceedings of SIGGRAPH*, pp. 131–144, ACM Press/Addison-Wesley Publishing Co.

[88] C. Loop (1987). *Smooth Spline Surfaces Based on Triangles.* Master's thesis, University of Utah, Department of Mathematics.

[89] C. Loop (2002). Bounded curvature triangle mesh subdivision with the convex hull property. *The Visual Computer* **18**:316–325.

[90] C. Loop (2002). Smooth ternary subdivision of triangle meshes. In *Curve and Surface Fitting: Saint-Malo 2002.*

[91] J. M. Lounsbery (1994). *Multiresolution Analysis for Surfaces of Arbitrary Topological Type.* Ph.D. thesis, University of Washington.

[92] S. Mallat (1998). *A wavelet tour of signal processing.* Academic Press.

[93] D. Marsh (1999). *Applied Geometry for Computer Graphics and CAD.* Springer-Verlag, London.

[94] W. S. Massey (1967). *Algebraic Topology: an Introduction.* Harcourt, Brace & World, Inc.

[95] M. Meyer, M. Desbrun, P. Schröder, A. H. Barr (2003). Discrete differential-geometry operators for triangulated 2-manifolds. In Springer (ed.), *Visualization and Mathematics III*, Mathematics and Visualization. Proceedings of VisMath '02 (Berlin).

[96] C. A. Micchelli, H. Prautzsch (1989). Uniform refinement of curves. *Linear Algebra and its Applications* **114/115**:841–870.

[97] F. Morán Burgos (2001). *Hierarchical Modelling of 3D Objects with Subdivision Surfaces.* Ph.D. thesis, Universidad Politécnica de Madrid.

[98] F. Morgan (1998). *Riemannian geometry. A beginner's guide.* A. K. Peters, Ltd., 2nd edn.

[99] P. Oswald (2002). Smoothness of nonlinear subdivision schemes. In *Intl. Conf. on Curves and Surfaces.*

[100] R. Pajarola, J. Rossignac (2000). Compressed progressive meshes. *IEEE Trans. on Visualization and Computer Graphics* **6**(1):79–93.

[101] J. Peters, U. Reif (1998). Analysis of algorithms generalizing B-spline subdivision. *SIAM Journal of Numerical Analysis* **35**(2):728–748.

[102] L. Piegl, W. Tiller (1997). *The NURBS Book.* Springer-Verlag, Berlin, Germany, 2nd edn.

[103] H. Prautzsch (1998). *Analysis of $\mathcal{C}^k$-subdivision surfaces at extraordinary points.* Tech. Rep. 98/4, Fakultät für Informatik, Universität Karlsruhe.

[104] H. Prautzsch (1998). Smoothness of subdivision surfaces at extraordinary points. *Advances in Computational Mathematics* **9**:377–389.

[105] H. Prautzsch, U. Reif (1999). Degree estimate for $\mathcal{C}^k$-piecewise polynomial subdivision surfaces. *Advances in Computational Mathematics* **10**(2):209–217.

[106] H. Prautzsch, G. Umlauf (1998). Improved triangular subdivision schemes. In F. E. Wolter, N. M. Patrikalakis (eds.), *Proceedings of te CGI'98*, pp. 626–632.

[107] H. Prautzsch, G. Umlauf (2000). A G^1 and G^2 subdivision scheme for triangular nets. *International Journal of Shape Modelling* **6**(1):21–35.

[108] M. Rabbani, R. Joshi (2002). An overview of the JPEG 2000 still image compression standard. *Signal Processing: Image Communication* **17**(1):3–48.

[109] U. Reif (1995). Some new results on subdivision algorithms for meshes of arbitrary topology. In C. K. Chui, L. L. Schumaker (eds.), *Wavelets and Multilevel Approximation*, vol. 2 of *Approximations and Decompositions*, pp. 367–374, World Scientific.

[110] U. Reif (1995). A unified approach to subdivision algorithms near extraordinary vertices. *Computer Aided Geometric Design* **12**(2):153–174.

[111] U. Reif, P. Schröder (2001). Curvature integrability of subdivision surfaces. *Advances in Computational Mathematics* **14**(2):157–174.

[112] R. F. Riesenfeld (1975). On Chaikin's algorithm. *Computer Graphics and Image Processing* **4**:304–310.

[113] O. Rioul (1992). Simple regularity criteria for subdivision schemes. *SIAM Journal of Mathematical Analysis* **23**(6):1544–1576.

[114] O. Rioul, T. Blu (1997). Simple regularity criteria for subdivision schemes. II. The rational case. Unpublished.

[115] J. Rossignac (1999). Edgebreaker: connectivity compression for triangle meshes. *IEEE Trans. on Visualization and Computer Graphics* **5**(1):47–61.

[116] V. Rvachev (1990). Compactly supported solutions of functional differential equations and their applications. *Russian Mathematical Surveys* **45**(1):87–120.

[117] A. Said, W. A. Pearlman (1996). A new and efficient image codec based on set partitioning in hierarchical trees. *IEEE Trans. on Circuits and Systems for Video Technology* **6**(3):243–250.

[118] D. Santa-Cruz (2003). *Compression of Parametric 3D models with NURBS.* Ph.D. thesis, École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland.

[119] P. Schröder (2002). Subdivision as a fundamental building block of digital geometry processing algorithms. *Journal of Computational and Applied Mathematics* **149**(1):207–219.

[120] J. Schweitzer (1996). *Analysis and Application of Subdivision Surfaces.* Ph.D. thesis, University of Washington.

[121] J. A. Sethian (1999). *Level Set Methods and Fast Marching Methods Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science.* Cambridge University Press.

[122] Sun Microsystems (2000). *The Java 3D API Specification, version 1.2.* Palo Alto, CA, USA. `http://java.sun.com/products/java-media/3D/forDevelopers/J3D_1_2_API/`.

[123] A. Szymczak, D. King, J. Rossignac (2001). An Edgebreaker-based efficient compression scheme for regular meshes. *Computational Geometry* **20**(1-2):53–68.

[124] G. Taubin, A. Guéziec, W. Horn, F. Lazarus (1998). Progressive forest split compression. In *Proceedinfs of SIGGRAPH'98*, pp. 123–132.

[125] G. Taubin, J. Rossignac (1998). Geometric compression through topological surgery. *ACM Transactions on Graphics* **17**(2):84–115.

[126] D. Taubman, E. Ordentlich, M. Weinberger, G. Seroussi (2002). Embedded block coding in JPEG 2000. *Signal Processing: Image Communication* **17**(1):73–84.

[127] D. S. Taubman, M. W. Marcellin (2002). *JPEG2000: image compression fundamentals, standards and practice.* Kluwer Academic Publishers.

[128] C. Touma, C. Gotsman (1998). Triangle mesh compression. In *Proc. of the 24th Conference on Graphics Interface (GI-98)*, pp. 26–34, Morgan Kaufmann, San Francisco, CA, USA.

[129] G. Turán (1984). On the succint representation of graphs. *Discrete Applied Mathematics* **8**:289–294.

[130] W. T. Tutte (1962). A census of planar triangulations. *Canadian Journal of Mathematics* **14**:21–38.

[131] G. Umlauf (2000). Analyzing the characteristic map of triangular subdivision schemes. *Constructive Approximation* **16**(1):145–155.

[132] M. Unser (1999). Splines: a perfect fit for signal and image processing. *IEEE Signal Processing Magazine* **16**(6):22–38.

[133] Y. Wang (2001). Subdivision schemes and refinement equations with nonnegative masks. *Journal of Approximation Theory* **113**:207–220.

[134] J. Warren (1995). Binary subdivision schemes for functions over irregular knot sequences. In M. Daehlen, T. Lyche, L. L. Schumaker (eds.), *Mathematical Methods for Curves and Surfaces*, pp. 543–562, Vanderbilt University Press.

[135] J. Warren (1995). *Subdivision methods for geometric design.* Tech. rep., Dept of Computer Science, Rice University.

[136] J. Warren, H. Weimer (2002). *Subdivision Methods for Geometric Design: a Constructive Approach.* Computer Graphics and Geometric Modeling. Morgan Kaufmann.

[137] Web3D Consortium (2001). *ISO/IEC 14772-1:1997/Amd.1:2002 Information Processing Systems - Computer Graphics The Virtual Reality Modeling Language Part 1 — Functional specification and UTF-8 encoding Amendment 1 — Enhanced interoperability.*

[138] Web3D Consortium (2002). *Information technology – Computer graphics and image processing – eXtensible 3D (X3D) – Part 1: Architecture and Base Components, Final Working Draft.* `http://www.web3d.org/TaskGroups/x3d/specification-milestone4/index.html`.

[139] I. H. Witten, R. M. Neal, J. G. Cleary (1987). Arithmetic coding for data compression. *Communications of the ACM* **30**(6):520–540.

[140] M. Woo, J. Neider, T. Davis, D. Shreiner (1999). *OpenGL Programming Guide.* Addison-Wesley, Reading, Massachusetts, 3rd edn.

[141] D. Zorin (1996). *$\mathcal{C}^k$-Continuity of Subdivision Surfaces.* Tech. Rep. 1996.CS-TR-96-23, Caltech.

[142] D. Zorin (1998). *Stationary Subdivision and Multiresolution Surface Representation.* Ph.D. thesis, California Institute of Technology, Pasadena, California.

[143] D. Zorin (2000). A method for analysis of $\mathcal{C}^1$-continuity of subdivision surfaces. *SIAM Journal of Numerical Analysis* **37**(5):1677–1708.

[144] D. Zorin (2000). Smoothness of stationary subdivision on irregular meshes. *Constructive Approximation* **16**(3):359–398.

[145] D. Zorin, P. Schröder, W. Sweldens (1996). Interpolation subdivision for meshes with arbitrary topology. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques, SIGGRAPH'96*, pp. 189–192, ACM Press, New Orleans, Louisiana.

Curriculum Vitæ

Name: Nicolas Aspert
Citizenship: French
Birthdate: December 23, 1975
Birthplace: Grenoble, France

Contact information

Address: Av. Vinet, 26
1004 Lausanne
Switzerland
Phone: +41 21 646 54 71
Email: Nicolas.Aspert@epfl.ch

Work experience

- ***February 1999 – present***: research assistant, Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland
  - Development of surface subdivision techniques, 3D model compression and watermarking algorithms.
  - Project management: participation in research project Neximage (image compression) and responsible for InterFace (audio-visual human-computer interface) and Certimark (watermarking algorithms design and evaluation) European research projects.
  - Teaching: coordinator for the "Image and video processing" course (2 semesters) exercises and lab. sessions.
  - Supervision: 4 diploma projects (4 to 6 months) and 2 undergraduate (4 months) students' projects
- ***July 1997 - September 1997***: Internship, Framatome Connectors International, Pontarlier (France)
  - Analysis of several electromagnetic compatibility measurements for RJ-45 network connectors.
  - Design and implementation of a coupling-attenuation measurement bench.

Education

- ***February 1999 – present***: PhD student in electrical engineering, Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland.
- ***September 1995 – September 1998***: *Diplôme d'Ingénieur Electricien ENSIEG* (Grenoble, France), equivalent of Master of Science in Electrical Engineering, obtained with honors. *Diplôme d'études approfondies* (Joseph Fourier University, Grenoble, France), with majors in signal, image and speech processing (required in France for PhD candidates) obtained at the same time, with honors.
- ***September 1997 – September 1998***: exchange student at Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland. Exchange fellowship awarded by the Rhône-Alpes region.
- ***September 1993 – June 1995***: *Classes préparatoires aux grandes écoles*, specialization in mathematics, Lycée Victor Hugo, Besançon France.

Skills

Languages

French:	mother tongue
English:	fluent oral and written
Spanish:	working knowledge

Computing

Operating systems:	Linux, Unix, Windows
Programming languages:	C, C++, Java, Perl, Unix shell
Other:	HTML, PHP, MySQL, LaTeX, Matlab, CVS

Publications

Journal papers

- Nicolas Aspert, Touradj Ebrahimi and Pierre Vandergheynst. Non-linear subdivision using local spherical coordinates. Computer-Aided Geometric Design, pp. 165–187, vol. 20, no. 3, June 2003.

Conference papers

- Nicolas Aspert, Diego Santa-Cruz and Touradj Ebrahimi. MESH: Measuring Errors between Surfaces using the Hausdorff distance. In Proc. of the IEEE International Conference in Multimedia and Expo (ICME) 2002, vol. 1, pp. 705–708, Lausanne, Switzerland, August 26–29, 2002.
- Nicolas Aspert, Elisa Drelie, Yannick Maret and Touradj Ebrahimi. Steganography for three-dimensional polygonal meshes. In Proc. of the 47th SPIE annual meeting, vol. 4790, pp. 210–219, Seattle, USA, 2002.

- Nicolas Aspert and Touradj Ebrahimi. Photo-realistic 3D model coding in MPEG-4. In Proc. of the IEEE International Conference in Multimedia and Expo (ICME) 2000, pp. 1111–1114, New York, USA, 2000.